SANDIA REPORT

SAND87–1305 • UC–814
Unlimited Release
Printed February 1993

Yucca Mountain Site Characterization Project

JAC3D—A Three-Dimensional Finite Element Computer Program for the Nonlinear Quasi-Static Response of Solids with the Conjugate Gradient Method

J. H. Biffle

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-76DP00789

134p.

"Prepared by Yucca Mountain Site Characterization Project (YMSCP) participants as part of the Civilian Radioactive Waste Management Program (CRWM). The YMSCP is managed by the Yucca Mountain Project Office of the U.S. Department of Energy, DOE Field Office, Nevada (DOE/NV). YMSCP work is sponsored by the Office of Geologic Repositories (OGR) of the DOE Office of Civilian Radioactive Waste Management (OCRWM)."

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A07
Microfiche copy: A01

SAND87-1305
Unlimited Release
Printed February 1993

Distribution
Category UC-814

JAC3D—A Three-Dimensional Finite Element Computer Program for the Nonlinear Quasi-Static Response of Solids with the Conjugate Gradient Method

J. H. Biffle
Computational Mechanics and Visualization Department
Sandia National Laboratories
Albuquerque, New Mexico 87185

Abstract

JAC3D is a three-dimensional finite element program designed to solve quasi-static nonlinear mechanics problems. A set of continuum equations describes the nonlinear mechanics involving large rotation and strain. A nonlinear conjugate gradient method is used to solve the equations. The method is implemented in a three-dimensional setting with various methods for accelerating convergence. Sliding interface logic is also implemented. An eight-node Lagrangian uniform strain element is used with hourglass stiffness to control the zero-energy modes. This report documents the elastic and isothermal elastic-plastic material model. Other material models, documented elsewhere, are also available. The program is vectorized for efficient performance on Cray computers. Sample problems described are the bending of a thin beam, the rotation of a unit cube, and the pressurization and thermal loading of a hollow sphere.

Acknowledgements

I would like to acknowledge several people who have been instrumental in the creation of JAC3D. In the beginning this work could not have been accomplished without the support and encouragement of Sam Key, my supervisor. Steve Burchett supplied numerous highly nonlinear problems to solve, which greatly helped define JAC3D's capabilities. Ray Krieg's contribution of the secant moduli derivation for the elastic-plastic model and Mark Blanford's restart algorithm for the conjugate gradient procedure are appreciated. Help in the preparation of the manuscript by Joe Jung and Mark Blanford is also greatly appreciated.

This report was prepared under the Yucca Mountain Project WBS number 1.2.4.2.3.1. The data in this report was developed subject to QA controls in QAGR S124231B, Revision 0, PCA 1.0, Task 1.1; the data is not qualified and is not to be used for licensing.

Contents

Figures

Tables

1. Introduction

1.1 Perspective

JAC3D is a finite element computer program for solving large deformation, temperature-dependent quasi-static mechanics problems in three dimensions. A nonlinear conjugate gradient technique (CG technique) is used to solve the governing nonlinear equations. A material model for elastic and isothermal elastic-plastic behavior with combined kinematic and isotropic hardening is described in this report. Other material models, documented elsewhere, are also available. An eight-node Lagrangian uniform-strain element is employed with hourglass stiffness to control the zero-energy modes.

JAC3D is very similar to the two-dimensional program JAC2D [1]. The JAC2D and JAC3D programs are the result of research to develop a reliable solution algorithm for solving quasi-static problems which executes efficiently on vector-processing computers. The nonlinear conjugate gradient method selected has proved to be very effective for solving three-dimensional problems.

1.2 Background

For the calculation of the nonlinear quasi-static response of solids, there is a need for efficient and reliable solution methods. In recent years, finite element nonlinear solutions to static problem have been obtained by using either a modified or unmodified Newton-Raphson method. Use of these stiffness approaches is troublesome because of the difficulty in deciding when to reformulate the stiffness matrix to keep the solution from diverging or to accelerate the convergence. On the opposite end of the spectrum of solution methods are indirect iterative methods, which do not involve a stiffness matrix.

The impetus to try indirect iterative solvers comes from several sources. First, a more robust method than the Newton-Raphson algorithm is needed to solve highly nonlinear problems involving geometric stiffening due to large deformations, stiffening and softening due to material response, and sudden changes in stiffness due to contact surface constraints. Second, there is a need to solve three-dimensional problems efficiently without a severe restriction on the number of elements that can be used because of hardware limitations in storing and retrieving the stiffness matrix from a magnetic disk.

Some of the motivation for trying indirect solution methods was obtained by observing the excellent results which explicit methods have produced in solving nonlinear transient dynamics problems. These methods have been very efficient in terms of computer resources. The data storage and code architecture for dynamics problems are very

similar to what is needed for indirect solution of statics problems. Examples of effective explicit dynamics codes include HONDO [2], WULFF [3], DYNA2D [4], DYNA3D [5], and recently, PRONTO2D [6] and PRONTO3D [7]. The research problem was to apply these concepts to an indirect solution method that is robust for nonlinear static problems.

In the early 1960s, indirect solution techniques such as successive overrelaxation, Gauss-Seidel, and Jacobi methods were tried on linear finite element equations. It was soon discovered that direct solution procedures (Gaussian elimination, for example) were much more efficient than indirect techniques if the equations were ordered in an efficient manner. However, only linear or mildly nonlinear problems were being solved at that time. Rashid reopened the question of whether to use iterative techniques for three-dimensional problems. His technique is discussed by Irons [8]. Indirect methods, if successful for two- and three-dimensional problems, could substantially reduce storage requirements and input-output operations when compared to the stiffness method. Moreover, the code could be highly vectorized, as demonstrated by the explicit dynamics codes. A reliable iterative method, even if expensive, is superior to a stiffness approach that does not reliably produce a solution on the first attempt.

After examining and trying various explicit techniques, the CG technique [9, 10, 11] was selected for solving highly nonlinear solid mechanics problems. These nonlinear effects include material nonlinearities and geometric nonlinearities due to large rotations, large strains, and surfaces that slide relative to one another. The CG technique was selected mainly for its reliability. In particular, convergence for a linear problem is guaranteed (with an infinite-precision machine) in N steps, where N is the number of unknowns in the problem. Also, various investigators in the field of linear programming and optimization are using the CG technique with success on very nonlinear problems [12, 13, 14]. Nonlinear versions of the CG technique are described by Daniel [15] and Bartels [16]. Several acceleration techniques for the linear CG methods are discussed in an article by Fletcher and Reeves [12]. The JAC3D implementation of the CG technique for solving nonlinear equations is discussed in Section 3.3.

In this document, the governing equations are formulated in the current configuration of the body, with particular attention being paid to the rotation of the stress tensor. The formulation is extremely convenient for the CG method because a stiffness matrix need not be calculated. Variational statements are then presented that allow a finite element representation of the equations of equilibrium.

1.3 Program Capabilities

The concepts noted above have been incorporated into the structural mechanics computer program JAC3D and combined with a variety of ancillary capabilities to result in a very versatile computer program.

1.3.1 Standard Geometry and Results File Format

As a member of the Sandia National Laboratories Engineering Analysis Code Access System (SEACAS) [17], JAC3D benefits from the rich computational analysis environment. Geometry and mesh information for the analysis is read from a file in the GENESIS format [18], which can be produced by a number of mesh generators and other preprocessors. Results are written to a file in the related EXODUS format [19], which is compatible with a suite of postprocessors and visualization aids.

1.3.2 Element Birth and Death

The program has the capability to add elements (element birth) and/or delete elements (element death) at selected times in the solution. This capability has proven to be an important feature, expecially for evaluating the residual stresses developed as a result of various manufacturing processes. For example, many electronic assemblies are built up by using a cascade of soldering steps. Two parts are joined with high-temperature solder, then a third part is added with a lower melt point solder, and so forth. Using the element birth capability, this manufacturing process can be realistically modeled, allowing new parts to appear at each step. In the same manner, changes in residual stress as the result of milling, drilling, or etching can be realistically modeled with the element death capability. As another example, mining operations can be modeled with the element death capability.

1.3.3 Material Models

At the present time, several nonlinear material constitutive models are incorporated in the program, with only one described here. The model is an isothermal elastic/plastic model with combined kinematic and isotropic hardening. The other models are documented separately, and more can be easily added. For a given problem, any or all of the material models which exist in the code can be used.

As an example, the elastic/plastic model is used extensively to describe the response of materials used in electronic assemblies. It has been successfully used to describe the behavior of ceramics, rigid polymers, solder at low temperature, and a host of other materials.

1.3.4 Initial Stress

Each material may be assigned an initial value for each component of stress in the reference configuration. The user may also specify a linear variation of stress in the z-coordinate direction. Initial stresses are typically specified to be in equilibrium with the initial boundary conditions. As an option, the user may request that the program calculate an initial equilibrium state before the first load step. In this case, two equilibrium

passes are made prior to beginning the load history; the displacements are zeroed out and the state variables reinitialized after each pass.

1.3.5 Kinematic Constraints

The geometric boundary conditions allow for nodal points to be rigidly fixed in space and time or to be defined to move in a specified time-dependent manner. This capability allows for realistic modeling of many quasi-static physical processes. For example, in electronics assemblies, connectors are often required. The mating of a connector pair can be described as press-fitting a contact pin into a housing. The requirements are that the contact force be sufficiently high to maintain electrical continuity; however, the stresses in the housing must remain linearly elastic so that the connector can be reliably used over and over. Time-dependent boundary conditions applied to the pin in conjunction with a contact surface definition between the pin and housing allow this problem to be modeled easily.

1.3.6 Loads

The program has the capability to apply a variety of mechanical time-dependent and/or time-constant loads to a model. These loads can be point loads, surface pressures, or body forces (arising from acceleration or electromagnetic fields). With these definitions, a great variety of mechanical loading applications can be modeled.

1.3.7 Thermal Input

The program has the capability to accept thermal input defining the temperature history of the structure. The temperature history can be obtained from a separate thermal analysis computer program, or generated with a user-supplied FORTRAN program. If the temperature history is uniform throughout the structure, it can be generated within JAC3D itself. Tracking the temperature history is important for a variety of applications. Residual thermal stresses can be developed during the manufacture of electronic assemblies; examples are soldering and brazing. The stresses are developed because of (1) the difference in the thermal expansion characteristics of the various materials in the assembly, or (2) the transient nonuniform temperature history. During use of the electronic assembly, the same problem arises as power is applied or removed. In addition, material response can vary as a function of temperature.

1.3.8 Contact Surfaces

The program can also model contacting surfaces. The contact surfaces can be fixed together, sliding without friction, or sliding with friction. They can be allowed to close or open as the solution dictates. This capability allows many physical processes, such as connector insertion, to be realistically modeled. The "fixed" contact surface has also

proven useful for grading element size, especially for the three-dimensional problem. This allows for parts of the structure to be very finely modeled to obtain the required resolution. The remainder of the structure, which is required to obtain the global response, can be modeled coarsely. These parts are joined by one or more fixed contact surfaces.

1.3.9 Restart

Finally, a capability to restart the solution is also incorporated. The restart can be used to change many of the problem parameters, thus allowing realistic physical processes to be modeled easily. For instance, stresses and deformations are generally developed in an electronic assembly due to its manufacture. Use environments impose additional stresses and deformations on the assembly. With the restart capability, an analysis of the manufacturing environment needs to be completed only one time. Various subsequent use environments can then be evaluated by restarting from this solution. The stress and deformation state existing in the restart file should be viewed as a set of equilibrium initial conditions with which to start a problem.

2. Governing Equations

This chapter gives the continuum mechanics concepts on which the development of the numerical algorithms in the following chapters is based. Boldface characters denote tensors. The order of the tensor may be determined from the context of the equation.

2.1 Kinematics

A material point in the reference configuration B_0 with position vector $\mathbf{X}$ occupies position $\mathbf{x}$ at time t in the deformed configuration B. Hence the notation $\mathbf{x} = \chi(\mathbf{X}, t)$. The motion from the original configuration to the deformed configuration shown in Figure 2.1 has a deformation gradient $\mathbf{F}$ given by

$$\mathbf{F} = \frac{\partial \mathbf{x}}{\partial \mathbf{X}} \, , \; |\mathbf{F}| > 0 \; . \tag{2.1}$$

Applying the polar decomposition theorem to $\mathbf{F}$,

$$\mathbf{F} = \mathbf{VR} = \mathbf{RU} \, , \tag{2.2}$$

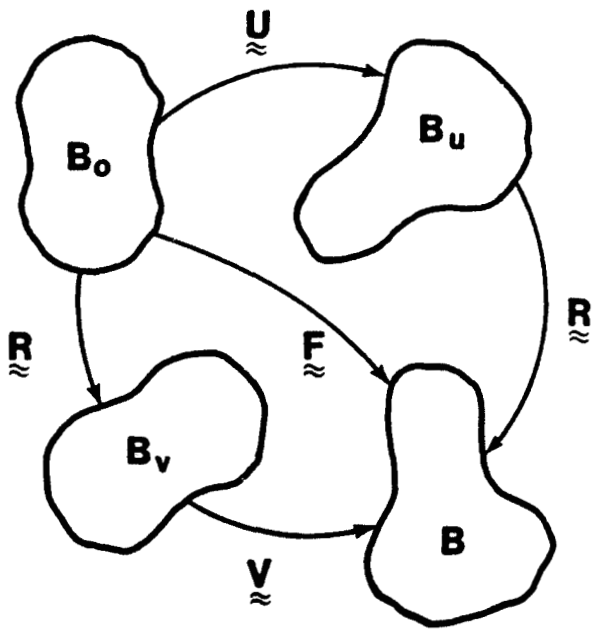


Figure 2.1. Original, Deformed, and Intermediate Configurations of a Body.

where $\mathbf{V}$ and $\mathbf{U}$ are the symmetric, positive definite left and right stretch tensors, respectively, and $\mathbf{R}$ is a proper orthogonal rotation tensor. Figure 2.1 illustrates the intermediate orientations defined by the two alternate decompositions of $\mathbf{F}$ defined by Equation 2.2. The determination of $\mathbf{R}$ follows from the work of Flanagan and Taylor [20]. The incremental algebraic algorithm to determine $\mathbf{R}$ is described in Section 4.2.

The velocity of the material point $\mathbf{X}$ is written as $\mathbf{v} = \dot{\mathbf{x}}$, where the superposed dot indicates time differentiation holding the material point fixed. The velocity gradient is denoted by $\mathbf{L}$ and may be expressed as

$$\mathbf{L} = \frac{\partial \mathbf{v}}{\partial \mathbf{x}} = \frac{\partial \mathbf{v}}{\partial \mathbf{X}} \frac{\partial \mathbf{X}}{\partial \mathbf{x}} = \dot{\mathbf{F}}\mathbf{F}^{-1} \ . \tag{2.3}$$

The velocity gradient can be written in terms of its symmetric ($\mathbf{D}$) and antisymmetric ($\mathbf{W}$) parts,

$$\mathbf{L} = \mathbf{D} + \mathbf{W} \ . \tag{2.4}$$

Using the right decomposition from Equation 2.2 in Equation 2.3 gives

$$\mathbf{L} = \dot{\mathbf{R}}\mathbf{R}^T + \mathbf{R}\dot{\mathbf{U}}\mathbf{U}^{-1}\mathbf{R}^T \ . \tag{2.5}$$

Dienes [21] denoted the first term on the right side of Equation 2.5 by $\mathbf{\Omega}$:

$$\mathbf{\Omega} = \dot{\mathbf{R}}\mathbf{R}^T \ . \tag{2.6}$$

Both $\mathbf{W}$ and $\mathbf{\Omega}$ are antisymmetric and represent a rate of rotation (or angular velocity) about some axes. In general, $\mathbf{\Omega} \neq \mathbf{W}$. The difference arises when the last term of Equation 2.5 is not symmetric. The symmetric part of $\dot{\mathbf{U}}\mathbf{U}^{-1}$ is the unrotated deformation rate tensor $\mathbf{d}$ as defined below (note that both $\dot{\mathbf{U}}$ and $\mathbf{U}^{-1}$ are symmetric).

$$\mathbf{d} = \frac{1}{2}(\dot{\mathbf{U}}\mathbf{U}^{-1} + \mathbf{U}^{-1}\dot{\mathbf{U}}) = \mathbf{R}^T\mathbf{D}\mathbf{R} \ . \tag{2.7}$$

There are two possible cases that can cause rotation of a material line element: rigid body rotation and shear. Since total shear vanishes along the axes of principal stretch, the rotation of these axes defines the total rigid body rotation of a material point.

With vector analysis it can be shown that Equation 2.6 represents the rate of rigid body rotation at a material point (as shown by Dienes). It can also be shown that $\mathbf{W}$ represents the rate of rotation of the principal axes of the rate of deformation $\mathbf{D}$. Since $\mathbf{D}$ and $\mathbf{W}$ have no sense of the history of deformation, they are not sufficient to define the rate of rotation in a finite deformation context.

Line elements where the rate of shear vanishes rotate solely due to rigid body rotations. These line elements are along the principal axes of $\dot{\mathbf{U}}$. A similar observation is applied below for using Dienes' expression for calculating $\mathbf{\Omega}$.

Using the left decomposition of Equation 2.2 in Equation 2.3 gives

$$\mathbf{L} = \dot{\mathbf{V}}\mathbf{V}^{-1} + \mathbf{V}\boldsymbol{\Omega}\mathbf{V}^{-1} \ . \tag{2.8}$$

Postmultiplying by $\mathbf{V}$ yields an expression that defines the decomposition of $\mathbf{L}$ into $\mathbf{V}$ and $\boldsymbol{\Omega}$:

$$\mathbf{L}\mathbf{V} = \dot{\mathbf{V}} + \mathbf{V}\boldsymbol{\Omega} \ . \tag{2.9}$$

When the dual vector of the above expression is taken, the symmetric $\dot{\mathbf{V}}$ vanishes to yield a set of three linear equations for the three independent components of $\boldsymbol{\Omega}$.

The antisymmetric part of a tensor may be expressed in terms of its dual vector and the permutation tensor e_{ijk}. Let us define the following dual vectors:

$$\omega_i = e_{ijk}\Omega_{jk} \tag{2.10}$$

$$w_i = e_{ijk}W_{jk} \ . \tag{2.11}$$

Using Equations 2.4, 2.10, and 2.11 in Equation 2.9 results in the expression that Dienes gave for determining $\boldsymbol{\Omega}$ from $\mathbf{W}$ and $\mathbf{V}$:

$$\boldsymbol{\omega} = \mathbf{w} - 2[\mathbf{V} - \mathbf{I}\,\mathrm{tr}\,\mathbf{V}]^{-1}\mathbf{z} \ , \tag{2.12}$$

where

$$z_i = e_{ijk}V_{jm}D_{mk} \ . \tag{2.13}$$

Since $\boldsymbol{\Omega} = \mathbf{W}$ if and only if the product $\mathbf{VD}$ is symmetric, then the principal axes of the deformation rate $\mathbf{D}$ coincide with the principal axes of the current stretch $\mathbf{V}$. Clearly, a pure rotation is a special case of this condition since $\mathbf{D}$, and consequently the z_i in Equation 2.13, vanish.

2.2 Stress and Strain Rates

The constitutive model architecture is posed in terms of the conventional Cauchy stress by adopting the approach of Johnson and Bammann [22] and defining a Cauchy stress in the unrotated configuration. More detail than is presented here is found in Flanagan and Taylor [20]. The "true" stress in the deformed configuration is denoted by $\mathbf{T}$. The Cauchy stress in the unrotated configuration is denoted by $\boldsymbol{\sigma}$. These two stress measures are related by

$$\boldsymbol{\sigma} = \mathbf{R}^T\mathbf{T}\mathbf{R} \ . \tag{2.14}$$

Each material point in the unrotated configuration has its own reference frame, which rotates in such a way that the deformation in this frame is a pure stretch. Then $\mathbf{T}$ is simply the tensor $\boldsymbol{\sigma}$ in the fixed global reference frame. The conjugate strain rate

measures to $\mathbf{T}$ and $\boldsymbol{\sigma}$ are $\mathbf{D}$ and $\mathbf{d}$, respectively. These strain rates were defined by Equations 2.4 and 2.7, respectively.

The principal of Material Frame Indifference (or objectivity) stipulates that a constitutive law must be insensitive to a change of reference frame [23]. This requires that only objective quantities may be used in a constitutive law. An objective quantity is one that transforms in the same manner as the energy conjugate stress and strain rate pair under a superposed rigid body motion. The fundamental advantage of the unrotated stress over the true stress is that the material derivative of $\boldsymbol{\sigma}$ is objective, whereas the material derivative of $\mathbf{T}$ is not.

A stress rate, called the Green-Naghdi rate by Johnson and Bammann, can be derived by transforming the rate of the unrotated Cauchy stress to the fixed global frame as follows:

$$\hat{\boldsymbol{\sigma}} = \mathbf{R}\dot{\boldsymbol{\sigma}}\mathbf{R}^T = \dot{\mathbf{T}} - \boldsymbol{\Omega}\mathbf{T} + \mathbf{T}\boldsymbol{\Omega} \ . \tag{2.15}$$

The Green-Naghdi rate is kinematically consistent with the rate of Cauchy stress. This statement means that $\hat{\boldsymbol{\sigma}}$ is identical to $\dot{\mathbf{T}}$ in the absence of rigid body rotations.

A distinct advantage of the unrotated reference frame is that all constitutive models are cast without regard to finite rotations. This greatly simplifies the numerical implementation of new constitutive models. The rotations of global state variables (e.g., stress and strain) are dealt with on a global level, which ensures that all constitutive models are consistent. Internal state variables (e.g., backstress) see no rotations whatsoever.

The drawback to working in the unrotated reference frame is that the rotation tensor $\mathbf{R}$ must be accurately determined. The incremental, algebraic algorithm to accomplish this task is described in Section 4.2.

2.3 Fundamental Equations

The quasi-static equations of motion for a body are

$$\nabla \cdot \mathbf{T} + \rho\mathbf{b} = \mathbf{0} \ , \tag{2.16}$$

where ρ is the weight density per unit volume and $\mathbf{b}$ is a specific (force per weight) body force vector.

The solution to Equation 2.16 is sought subject to the boundary conditions

$$\mathbf{u} = \mathbf{f}(t) \text{ on } S_u \ , \tag{2.17}$$

where S_u represents the portion of the boundary on which kinematic quantities are specified (displacement and velocity). In addition to satisfying the kinematic boundary conditions given by Equation 2.17, the traction boundary conditions must be satisfied as

$$\mathbf{T} \cdot \mathbf{n} = \mathbf{s}(t) \text{ on } S_T \ , \tag{2.18}$$

where S_T represents the portion of the boundary on which tractions are specified. The boundary of the body is given by the union of S_u and S_T, and for a valid mechanics problem, S_u and S_T have a null intersection.

3. Numerical Solution Procedure

The solution to the quasi-static problem, described by Equations 2.16, 2.17 and 2.18, is calculated at discrete points in time by obtaining a minimum of a functional Π, which represents the power input to the body. The nonlinear conjugate gradient method is used to minimize the functional, and the finite element technique is employed to discretize the problem geometrically. The reader should be familiar with the finite element method. If not, numerous texts on the method, such as Bathe and Wilson [24], can be consulted.

3.1 Time Integration Procedure

Equations 2.16, 2.17 and 2.18 describe a quasi-static theory in which velocities are retained but the time rates of velocities are neglected. Some quasi-static mechanical processes depend on real time, such as those involving viscoelastic and creeping materials. Others, such as those which involve elastic or elastic/plastic materials, proceed independently of the amount of time used in the process. In any event, an incremental solution in time is used to describe the nonlinear process. For the solution increment going from time t_n to t_{n+1}, an interval of time Δt is used

$$\Delta t = t_{n+1} - t_n \ ,$$

where n is called the *time step* (or *load step*) number.

3.2 The Functional or Objective Function

The goal is to obtain a solution at discrete times by finding the minimum of a nonlinear functional which represents the fundamental equations. We begin in defining the functional or objective function by writing the power input to the body (which is zero for the quasi-static problem) as

$$P_{\text{input}} = \int_S s_i \dot{u}_i \, dS + \int_V \rho b_i \dot{u}_i \, dV \ , \tag{3.1}$$

where S denotes the surface and V is the volume of the body in the deformed configuration. A discussion of the power input to the body can be found in Malvern [25].

Using Equation 2.18, the surface integral in Equation 3.1 can be transformed into

$$P_{\text{input}} = \int_V [\dot{u}_i (T_{ji,j} + \rho b_i) + T_{ij} \dot{u}_{i,j}] \, dV \ . \tag{3.2}$$

A functional Π is now defined by equating Equations 3.1 and 3.2 as

$$\Pi = \int_V [\dot{u}_i T_{ji,j} + T_{ji}\dot{u}_{i,j}]\, dV - \int_S s_i \dot{u}_i\, dS = 0 \ . \tag{3.3}$$

If the equilibrium Equations 2.16 are substituted for $T_{ji,i}$ in Equation 3.3, the second term is integrated by parts, and the first variation is taken, the result is

$$\delta\Pi = \int_V \delta\dot{u}_i(-\rho b_i - T_{ji,j})\, dV - \int_{S_T} \delta\dot{u}_i(\hat{s}_i - s_i)\, dS \ . \tag{3.4}$$

The Euler equations are the traction boundary conditions (Equations 2.18) and the statement of equilibrium (Equations 2.16). If, in Equation 3.4, the term involving $T_{ji,i}$ is integrated by parts, the following first variation is obtained:

$$\delta\Pi = \int_V \delta\dot{u}_i(-\rho b_i)\, dV + \int_V \delta\dot{u}_{i,j} T_{ij}\, dV - \int_{S_T} \delta\dot{u}_i s_i\, dS = 0 \ . \tag{3.5}$$

The minimum of the functional at a specified time will be found with the use of the nonlinear conjugate gradient procedure. Equation 3.5 is used to determine the gradient of the objective function (i.e., the residual forces in the body) at each iteration, and the finite element method is used to discretize the body. Since $\delta\dot{u}_i$ represents an arbitrary virtual velocity field, Equation 3.5 is rewritten (with the use of the traction boundary condition Equation 2.18) as a summation of the contributions of force from each finite element to obtain

$$R = \sum_e \left[\int_{V_e} T_{ij}\delta\dot{u}_{i,j}\, dV - \int_{V_e} \rho b_i \delta\dot{u}_i\, dV - \int_{S_e} T_{ij} n_j \delta\dot{u}_i\, dA \right] \ . \tag{3.6}$$

The summation symbol represents the assembly of element force vectors into a global nodal force array. It is assumed that the reader understands the details of this assembly. In general, while iterating towards a solution within a load step, the value of the *residual vector* R in Equation 3.6 will not be zero. In fact, convergence is defined by a measure of how close R is to zero.

3.3 Conjugate Gradient Algorithm

For a quasi-static time step, a trial solution of components of the velocity vector is substituted into the set of nonlinear Equations 3.6 and the residual vector (the gradient of the functional Π) is obtained:

$$R(\dot{u}) = \delta\Pi(\dot{u}) \ . \tag{3.7}$$

In the indirect iterative solution procedure, a set of velocity components is sought that will make the residual vector zero or acceptably small. The conjugate gradient method

is used to efficiently obtain directions in which to search for the velocity solution. Using a form of the conjugate gradient method obtained by combining a linear preconditioned version [11] and a nonlinear version [16], the iterative process is started by assuming a vector of velocity components at the nodes of the finite element mesh, $\dot{u}_j$, with j denoting the iteration number. The residual vector, the gradient of the functional, becomes

$$R_j = R(\dot{u}_j) \ . \tag{3.8}$$

A preconditioning matrix M (the diagonal of the linear stiffness matrix) is introduced to define a generalized gradient vector Z as follows:

$$M_n Z_j = R_j \ . \tag{3.9}$$

The conditioning is helpful when the body contains materials of different stiffness or elements of widely varying sizes.

If $j = 0$, the initial search direction is the negative of the gradient, the steepest descent direction P_0:

$$P_0 = -Z_0 = -M_n^{-1} R_0 \tag{3.10}$$

Subsequently, for $j > 0$, search directions that are conjugate to the previous direction are chosen as follows:

$$P_j = -Z_j + \beta_j P_{j-1} \ , \tag{3.11}$$

where β_j has the value

$$\beta_j = \frac{Z_j^T M_n (Z_j - Z_{j-1})}{Z_{j-1}^T M_n Z_{j-1}} \tag{3.12}$$

Equation 3.12 is a generalization of a method known as the Polak-Ribiére algorithm, as discussed by Powell [14]. The variables $\dot{u}$ are then updated by searching for the least value of $\Pi(\dot{u})$ from $\dot{u}$ along the direction P_j. Therefore

$$\dot{u}_{j+1} = \dot{u}_j + \alpha_j P_j \tag{3.13}$$

where α_j is the value that minimizes the function of one variable. (The process of finding α_j is known also as a line search.) Therefore

$$R(\alpha_j) = \delta\Pi(\dot{u}_j + \alpha_j P_j) \tag{3.14}$$

After calculating Equation 3.13, if the residual R is not acceptably small, another iteration is begun. Efficient use of the conjugate gradient method greatly depends upon the cost of the line search (calculating α).

3.4 Gradient Calculations

The cost of solving problems with the conjugate gradient method is dominated by the cost of gradient calculations. There are two places in the conjugate gradient procedure where gradient calculations are needed. The first occurs in computing Equation 3.8, the residual force vector, for each iteration. The second set of gradient calculations is required when performing the line search represented by Equation 3.14.

3.4.1 Residual Force

The calculation of the residual force vector or gradient of the objective function, Equation 3.6, is accomplished by calculating contributions of force from several sources. Forces are caused by the state of internal stress, artificial forces to stabilize singular modes of elements (hourglass modes), external applied tractions, internal body forces, and externally applied point loads. The specific method of calculating these forces varies with the element type, as described in Chapter 4.

3.4.2 Line Search

It is necessary to find α_j which will minimize Equation 3.14. Equation 3.14 is nonlinear in α_j, and it is often solved iteratively for α_j using Newton's method. However, if the problem is highly nonlinear, Newton's method can take many iterations. This requires many residual calculations which will dominate the cost of an analysis. Following Bartels and Daniel [16], the minimization solution can effectively be approximated by one step of Newton's method starting with $\alpha_j = 0$. The Newton process will result in the following expression for α_j:

$$\alpha_j = \frac{Z_j^T M_n Z_j}{P_j^T R_{P_j}} \,. \tag{3.15}$$

The term R_{P_j} represents a residual calculation with the P_j vector substituted for the velocity vector. The material constitutive model is required to supply a secant modulus array for use in calculating R_{P_j}. If the material model cannot supply a secant modulus, then it is approximated using the elastic moduli of the material. The use of the secant modulus and the single-step Newton's method to perform the line search has proven to be very economical when the material responds according to the elastic/plastic constitutive law. If the problem is linear, both geometrically and in material response, then the single-step Newton's method performs an exact line search for α_j.

3.5 Restarting the Algorithm

The algorithm described in Section 3.3 draws its strength from finding new search directions which are orthogonal (or conjugate) to those already taken. In some highly nonlinear problems, however, this can become a drawback.

The problem comes when the nonlinear functional $\delta\Pi$ which depends on u as well as on $\dot{u}$ has changed enough from $\dot{u}_0$ to $\dot{u}_k$, or when roundoff error or other approximations have accumulated to the point that the solution that minimizes Equation 3.8 may have substantial components in directions that have already been searched. This becomes apparent when no further reduction in the size of the residual is obtained even after a great many iterations, or when the residuals begin growing very large ("blowing up"). In such cases one must start the algorithm over again using a new (perhaps closer) initial guess.

In such cases, the program can pick a new initial guess "on the fly," selecting as its new guess the vector $\dot{u}_{j_{\min}}$ which has produced the smallest residual $R_{j_{\min}}$ so far in the current load step. Three parameters governing this strategy may be adjusted using the following input record:

CGRESET LIMITS *itstrt, itrset, tolfac*

First, the new guess $\dot{u}_{j_{\min}}$ must differ enough from the original guess $\dot{u}_0$ that it will produce better results. The first parameter, *itstrt*, specifies how many iterations to wait before looking for a minimum residual. i.e., it is required that $j_{\min} > itstrt$. The default value is one percent of the number of degrees of freedom: for a 1000-node problem, *itstrt* would default to 30.

The next problem is to decide *when* to give up on the current CG iteration series and try a new guess. Currently two situations are targeted: (1) many iterations with no further reduction in residual size, and (2) a large increase in residual size, indicating divergence. The second CGRESET LIMITS parameter, *itrset*, specifies the number of iterations to allow between finding a minimum and restarting the CG algorithm. The default value of *itrset* is half the number of degrees of freedom. The third parameter, *tolfac*, defines how much growth in the residual norm indicates divergence. Its default is 1000, meaning that if the norm of the residual grows three orders of magnitude from its minimum value so far, restart the algorithm. Both these defaults are intentionally loose, so that the restart logic will provide a safety net without interfering with a properly-functioning CG solution.

In the extreme, setting *itstrt* to 0 and *itrset* to 1 results in restarting the CG algorithm every iteration. This reduces the iteration scheme to the steepest descent method, always moving the solution in the direction of the current residual. Convergence of the steepest descent method is often much slower than that of the conjugate gradient method. The CG algorithm needs a "long leash" to function properly; if the CGRESET LIMITS are set too tightly, the convergence rate will suffer accordingly, approaching that of the steepest descent method. For a problem that is just not converging very well, watching the progress of the iterations with ITERATION PRINT turned on should indicate what CGRESET LIMITS may be most helpful. It may be beneficial to reduce the size of the load step and/or adjust the TRIAL VELOCITY FACTOR as well.

3.6 Convergence

Global convergence at the end of a time step is defined to have taken place when any of the following inequalities is satisfied:

$$\frac{\|R_j\|}{\|F_n\|} \leq \texttt{TOLR}\,, \tag{3.16}$$

$$\|R_j\| < \texttt{RESIDF}\,, \tag{3.17}$$

or

$$\frac{\|\dot{u}_j\| - \|\dot{u}_{j-1}\|}{\|\dot{u}_j\|} < \texttt{TOLU} \ . \tag{3.18}$$

$\|\cdot\|$ denotes the L_2 norm of a vector. In Equation 3.16, F_n is a vector containing the applied tractions, body forces (gravity forces), thermal forces, and the reactions at nodes where nonzero displacement boundary conditions are applied. Equation 3.17 provides a convergence check for situations where the applied loads are small or nonexistent (i.e., when unloading to a zero load). Equation 3.18 is used to measure the change in the velocity vector due to one conjugate gradient iteration. Its main purpose is stop the solution attempt if little progress is being made towards a solution. The velocity convergence criterion should not be relied upon as a statement that the problem is to a state which is close to equilibrium. However, Equations 3.16 and 3.17 are good measures of how close the problem is to a state of equilibrium. The program will terminate iterations for the load step if any of these conditions is satisfied. The default tolerances for Equations 3.16 and 3.18 are 1.0×10^{-3} and 1.0×10^{-12}, respectively, whereas RESIDF in Equation 3.17 defaults to zero.

If none of the above conditions is satisfied within the user-supplied MAXIMUM ITERATIONS, the program will first go back to the iterate $\dot{u}_{j_{\min}}$ that produced the smallest residual during the load step iterations. If the relative size of the corresponding residual $R_{j_{\min}}$ is less than the user-specified MAXIMUM TOLERANCE, then $\dot{u}_{j_{\min}}$ is accepted and the program will proceed to the next load step. If not, $\dot{u}_{j_{\min}}$ is written to the plot file and the analysis is terminated. The default value for MAXIMUM ITERATIONS is the number of degrees of freedom, while MAXIMUM TOLERANCE defaults to zero.

4. Finite Element Calculations

To define an isoparametric finite element, the spatial coordinates x_i are related to the nodal coordinates x_{iI} through the isoparametric shape functions ϕ_I as follows:

$$x_i = x_{iI}\phi_I(\xi,\eta,\zeta) \ . \tag{4.1}$$

In accordance with indicial notation convention, repeated subscripts imply summation over the range of that subscript. The lowercase subscripts have a range of three corresponding to the spatial coordinate directions. Uppercase subscripts have a range corresponding to the number of element nodes.

The same shape functions are used to define the element displacement field in terms of the nodal displacements u_{iI}:

$$u_i = u_{iI}\phi_I \ . \tag{4.2}$$

Since the same shape functions apply to both spatial coordinates and displacements, their material derivative (represented by a superposed dot) must vanish. Hence, the velocity field may be given by

$$\dot{u}_i = \dot{u}_{iI}\phi_I \ . \tag{4.3}$$

The velocity gradient tensor $\mathbf{L}$ is defined in terms of nodal velocities as

$$L_{ij} = \dot{u}_{i,j} = \dot{u}_{iI}\phi_{I,j} \ . \tag{4.4}$$

By convention, a comma preceding a lowercase subscript denotes differentiation with respect to the spatial coordinates (e.g., $\dot{u}_{i,j}$ denotes $\partial\dot{u}_i/\partial x_j$).

4.1 Element Library

Two element types are currently included in JAC3D: one continuum element and one structural element. The continuum element is an eight-node uniform strain element. The structural element is a two-node elastic truss.

4.1.1 Eight-Node Uniform Strain Element

The eight-node three-dimensional isoparametric element is widely used in computational mechanics. The determination of optimal integration schemes for this element, however, presents a difficult dilemma. A one-point integration of the element underintegrates the element, resulting in a rank deficiency that manifests itself in spurious zero energy modes, commonly referred to as hourglass modes. A two-by-two-by-two integration of the element over-integrates the element and can lead to serious problems of

element locking in fully-plastic and incompressible problems. The eight-point integration also carries a tremendous computational cost penalty compared to the one-point rule. In JAC3D a one-point integration of the element is used and implemented with an hour-glass control scheme to eliminate the spurious modes. The development presented below follows directly from Flanagan and Belytschko [26].

The 3-D isoparametric shape functions map the unit cube in ξ_i-space (ξ_i is written explicitly as (ξ, η, ζ)) to a general hexahedron in x_i-space, as shown in Figure 4.1. The unit square is centered at the origin in ξ_i space so that the shape functions may be conveniently expanded in terms of an orthogonal set of base vectors, given in Table 4.1, as follows:

$$\phi_I = \frac{1}{8}\Sigma_I + \frac{1}{4}\xi\Lambda_{1I} + \frac{1}{4}\eta\Lambda_{2I} + \frac{1}{4}\zeta\Lambda_{3I} + \frac{1}{2}\eta\zeta\Gamma_{1I} + \frac{1}{2}\zeta\xi\Gamma_{2I} + \frac{1}{2}\xi\eta\Gamma_{3I} + \frac{1}{2}\xi\eta\zeta\Gamma_{4I} \quad (4.5)$$

Note that the notation follows that used by Flanagan and Belytschko. In their work the ξ_i range from $-\frac{1}{2}$ to $\frac{1}{2}$.

The above vectors represent the displacement modes of a unit cube. The first vector, Σ_I, accounts for rigid body translation. Σ is called the summation vector because it may be employed in indicial notation to represent the algebraic sum of vector components.

The linear base vectors denoted by Λ_{iI} may be readily combined to define three uniform normal strains and three rigid body rotation modes for the unit cube. The Λ_{iI} are referred to as the volumetric base vectors since, as is illustrated below, they are the only base vectors which appear in the element volume expression.

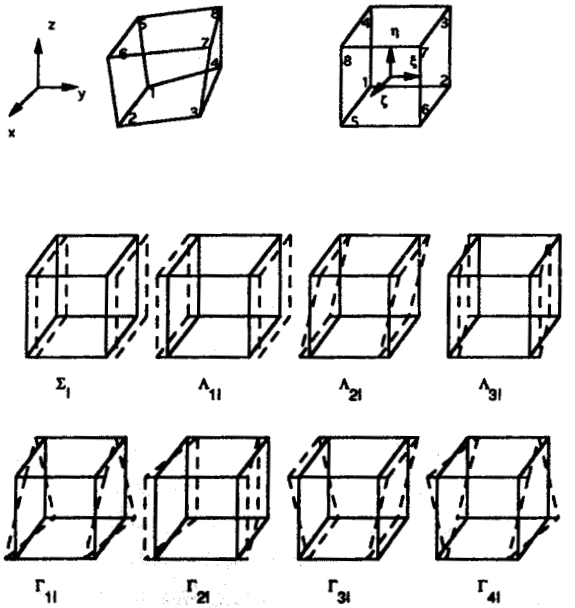


Figure 4.1. Mode Shapes for the Eight-Node Constant Strain Hexahedral Element.

Table 4.1. Orthogonal Set of Base Vectors

Node	ξ	η	ζ	Σ_I	Λ_{1I}	Λ_{2I}	Λ_{3I}	Γ_{1I}	Γ_{2I}	Γ_{3I}	Γ_{4I}
1	−.5	−.5	−.5	1	−1	−1	−1	1	1	1	−1
2	.5	−.5	−.5	1	1	−1	−1	1	−1	−1	1
3	.5	.5	−.5	1	1	1	−1	−1	−1	1	−1
4	−.5	.5	−.5	1	−1	1	−1	−1	1	−1	1
5	−.5	−.5	.5	1	−1	−1	1	−1	−1	1	1
6	.5	−.5	.5	1	1	−1	1	−1	1	−1	−1
7	.5	.5	.5	1	1	1	1	1	1	1	1
8	−.5	.5	.5	1	−1	1	1	1	−1	−1	−1

The last four vectors denoted by $\Gamma_{\alpha I}$ (Greek subscripts have a range of four) give rise to linear strain modes that are neglected in the single point integration. These vectors define the hourglass patterns for a unit cube. Hence, the $\Gamma_{\alpha I}$ are referred to as the hourglass base vectors. The displacement modes represented by the vectors in Table 4.1 are also shown in Figure 4.1.

The first integral in Equation 3.6 is used to define the element internal force vector f_{iI} as

$$\delta\dot{u}_{iI} f_{iI} = \int_{V_e} T_{ij}\delta\dot{u}_{i,j}\, dV \ . \tag{4.6}$$

The second and third integrals define the external force vector.

One-point integration is performed by neglecting the nonlinear portion of the element velocity field, thereby considering a state of uniform strain and stress. The preceding expression is approximated by

$$f_{iI} = \bar{T}_{ij} \int_{V_e} \phi_{I,j}\, dV \ , \tag{4.7}$$

where the arbitrary virtual velocities are eliminated, and $\bar{T}_{ij}$ represents the assumed uniform stress field, which will be referred to as the mean stress tensor. Neglecting the nonlinear velocities results in the mean stresses depending only on the mean strains. Mean kinematic quantities are defined by integrating over the element as follows:

$$\dot{\bar{u}}_{i,j} = \frac{1}{V} \int_{V_e} \dot{u}_{i,j}\, dV \ . \tag{4.8}$$

The discrete gradient operator is defined as

$$B_{iI} = \int_{V_e} \phi_{I,i}\, dV \ . \tag{4.9}$$

The mean velocity gradient, applying Equation 4.4, is given by

$$\dot{\bar{u}}_{i,j} = \frac{1}{V}\dot{u}_{iI}B_{jI} \ . \tag{4.10}$$

Combining Equations 4.7 and 4.9, the nodal forces are expressed by

$$f_{iI} = \bar{T}_{ij}B_{jI} \ . \tag{4.11}$$

Computing nodal forces with this integration scheme requires evaluation of the gradient operator and the element volume. These two tasks are linked since

$$x_{i,j} = \delta_{ij} \ , \tag{4.12}$$

where δ_{ij} is the Kroneker delta. Equations 4.1, 4.9, and 4.12 yield

$$x_{iI}B_{jI} = \int_{V_e} (x_{iI}\phi_I)_{,j} \, dV = V\delta_{ij} \ . \tag{4.13}$$

Consequently, the gradient operator may be expressed by

$$B_{iI} = \frac{\partial V}{\partial x_{iI}} \ . \tag{4.14}$$

To integrate the element volume in closed form, the Jacobian of the isoparametric transformation is used to transform to an integral over the unit cube:

$$V = \int_V dV = \int_{-\frac{1}{2}}^{+\frac{1}{2}} \int_{-\frac{1}{2}}^{+\frac{1}{2}} \int_{-\frac{1}{2}}^{+\frac{1}{2}} J \, d\zeta \, d\eta \, d\xi \ . \tag{4.15}$$

The Jacobian is given in terms of the alternator e_{ijk} as

$$J = e_{ijk}\frac{\partial x}{\partial \xi_i}\frac{\partial y}{\partial \eta_j}\frac{\partial z}{\partial \zeta_k} \ . \tag{4.16}$$

Therefore, Equation 4.15 can be written as

$$V = x_I y_J z_K C_{IJK} \ , \tag{4.17}$$

where

$$C_{IJK} = e_{ijk} \int_{-\frac{1}{2}}^{+\frac{1}{2}} \int_{-\frac{1}{2}}^{+\frac{1}{2}} \int_{-\frac{1}{2}}^{+\frac{1}{2}} \frac{\partial \phi_I}{\partial \xi_i}\frac{\partial \phi_J}{\partial \eta_j}\frac{\partial \phi_K}{\partial \zeta_k} \, d\zeta \, d\eta \, d\xi \ . \tag{4.18}$$

Observe that the coefficient array C_{IJK} is identical for all hexahedrons. Furthermore, it possesses the alternator properties as given below:

$$C_{IJK} = C_{JKI} = C_{KIJ} = -C_{IKJ} = -C_{JIK} = -C_{KJI} \ . \tag{4.19}$$

Therefore, applying Equations 4.14 and 4.19 to 4.17 yields the following form for evaluating the B-matrix:

$$B_{iI} = \begin{bmatrix} y_J z_K \\ z_J x_K \\ x_J y_K \end{bmatrix} C_{IJK} \ . \tag{4.20}$$

In light of Equation 4.5, it is evident that evaluating each component of C_{IJK} involves integrating a polynomial that is at most bi-quadratic. However, since integration is over a symmetric region, any term with a linear dependence will vanish. The only terms to survive the integration will be the constant, square, double square, and triple square terms. Furthermore, the alternator properties cause half of these remaining terms to drop out. The resulting expression for C_{IJK} is

$$C_{IJK} = \frac{1}{192} e_{ijk} \left(3\Lambda_{iI}\Lambda_{jJ}\Lambda_{kK} + \Lambda_{iI}\Gamma_{kJ}\Gamma_{jK} + \Gamma_{kI}\Lambda_{jJ}\Gamma_{iK} + \Gamma_{jI}\Gamma_{iJ}\Lambda_{kK}\right) \tag{4.21}$$

The above expression is evaluated using Table 4.1, after which practical formulas for computing the B-matrix and volume are developed. Since C_{IJK} has the alternator properties given in Equation 4.19, only 56 distinct nonzero terms (combinations of eight nodes taken three at a time) are possible. However, the volume must be independent of the selection of node 1, which implies that C_{IJK} is invariant if the nodes are permuted according to Table 4.2. Consequently, only 21 terms (combinations of seven nodes taken two at a time) may be independent. Furthermore, once node 1 is selected, three orientations of the node numbering system are possible, as given by the permutations in Table 4.3. Therefore, only seven terms of C_{IJK} need be evaluated.

Seven independent terms of C_{IJK} are listed in Table 4.4. These terms may be evaluated via Equation 4.21 and Table 4.1. Only three of these seven terms do not vanish, as indicated in Table 4.4. All other nonzero terms of C_{IJK} are found by permuting the nodes according to Table 4.2 and using the alternator properties of Equation 4.19. Alternatively, the nonzero terms may be generated by applying antisymmetry ($C_{IJK} = -C_{IKJ}$) to Table 4.4, then permuting according to Tables 4.3 and 4.2, successively. The latter scheme straightforwardly results in formulas for computing the B-matrix.

The first term of B_{iI} is expressed as

$$\begin{aligned} B_{11} \ = \ & \frac{1}{12}\Big[y_2[(z_6 - z_3) - (z_4 - z_5)] + y_3(z_2 - z_4) + y_4[(z_3 - z_8) - (z_5 - z_2)] \\ & + y_5[(z_8 - z_6) - (z_2 - z_4)] + y_6(z_5 - z_2) + y_8(z_4 - z_5)\Big] \end{aligned} \tag{4.22}$$

Other terms of B_{iI} are evaluated using the same formula after permuting the nodes according to Table 4.2 and, subsequently, permuting the coordinate axes according to Table 4.5. The element volume is most easily computed by contracting the B-matrix and nodal coordinates as per Equation 4.13.

Table 4.2. Nodal Permutations

1	2	3	4	5	6	7	8
2	3	4	1	6	7	8	5
3	4	1	2	7	8	5	6
4	1	2	3	8	5	6	7
5	8	7	6	1	4	3	2
6	5	8	7	2	1	4	3
7	6	5	8	3	2	1	4
8	7	6	5	4	3	2	1

Table 4.3. Three Possible Orientations of Node Numbering

1	2	3	4	5	6	7	8
1	4	8	5	2	3	7	6
1	5	6	2	4	8	7	3

Table 4.4. Nonzero Terms Generated by Applying Asymmetry

I	J	K	C_{IJK}
1	2	3	$-\frac{1}{12}$
1	2	5	$+\frac{1}{12}$
1	2	6	$+\frac{1}{12}$
1	2	7	0
1	2	8	0
1	3	5	0
1	3	6	0

Table 4.5. Coordinate Axes Permutations

1	2	3
2	3	1
3	1	2

Hourglass Control Algorithm

The mean stress-strain formulation of the uniform strain element considers only a fully linear velocity field. The remaining portion of the nodal velocity field is the so-called hourglass field. Excitation of these modes may lead to severe, unresisted mesh distortion. The hourglass control algorithm described here is taken directly from Flanagan and Belytschko [26]. The method isolates the hourglass modes so that they may be treated independently of the rigid body and uniform strain modes.

A fully linear velocity field for the hexahedron can be described by

$$\dot{u}_i^{\text{lin}} = \dot{\bar{u}}_i + \dot{\bar{u}}_{i,j}(x_j - \bar{x}_j) \ . \tag{4.23}$$

The mean coordinates $\bar{x}_i$ correspond to the center of the element and are defined as

$$\bar{x}_i = \frac{1}{8} x_{iI} \Sigma_I \ . \tag{4.24}$$

The mean translational velocity is similarly defined by

$$\dot{\bar{u}}_i = \frac{1}{8} \dot{u}_{iI} \Sigma_I \ . \tag{4.25}$$

The linear portion of the nodal velocity field may be expressed by specializing Equation 4.23 to the nodes as follows:

$$\dot{u}_{iI}^{\text{lin}} = \dot{\bar{u}}_i \Sigma_I + \dot{\bar{u}}_{i,j}(x_{jI} - \bar{x}_j \Sigma_I) \ , \tag{4.26}$$

where Σ_I is used to maintain consistent index notation and indicates that $\dot{\bar{u}}_i$ and $\bar{x}_j$ are independent of position within the element. From Equations 4.10 and 4.26, and the orthogonality of the base vectors, it follows that

$$\dot{u}_{iI} \Sigma_I = \dot{u}_{iI}^{\text{lin}} \Sigma_I = 8 \dot{\bar{u}}_i \tag{4.27}$$

and

$$\dot{u}_{iI} B_{jI} = \dot{u}_{iI}^{\text{lin}} B_{jI} = V \dot{\bar{u}}_{i,j} \ . \tag{4.28}$$

The hourglass field $\dot{u}_{iI}^{\text{hg}}$ may now be defined by removing the linear portion of the nodal velocity field:

$$\dot{u}_{iI}^{\text{hg}} = \dot{u}_{iI} - \dot{u}_{iI}^{\text{lin}} \tag{4.29}$$

Equations 4.27 through 4.29 prove that Σ_I and B_{jI} are orthogonal to the hourglass field:

$$\dot{u}_{iI}^{\text{hg}} \Sigma_I = 0 \tag{4.30}$$

$$\dot{u}_{iI}^{\text{hg}} B_{jI} = 0 \ . \tag{4.31}$$

Furthermore, it can be shown that the B-matrix is a linear combination of the volumetric base vectors Λ_{iI}, so Equation 4.31 can be written as

$$\dot{u}_{iI}^{\text{hg}} \Lambda_{iI} = 0 \ . \tag{4.32}$$

Equations 4.30 and 4.32 show that the hourglass field is orthogonal to all the base vectors in Table 4.1 except the hourglass base vectors. Therefore, $\dot{u}_{iI}^{\mathrm{hg}}$ may be expanded as a linear combination of the hourglass base vectors as follows:

$$\dot{u}_{iI}^{\mathrm{hg}} = \frac{1}{\sqrt{8}} \dot{q}_{i\alpha} \Gamma_{\alpha I} \ . \tag{4.33}$$

The hourglass nodal velocities are represented by $\dot{q}_{i\alpha}$ above (the leading constant is added to normalize $\Gamma_{\alpha I}$). The hourglass shape vector $\gamma_{\alpha I}$ is defined such that

$$\dot{q}_{i\alpha} = \frac{1}{\sqrt{8}} \dot{u}_{iI} \gamma_{\alpha I} \ . \tag{4.34}$$

By substituting Equations 4.26, 4.29, and 4.34 into 4.33, then multiplying by $\Gamma_{\alpha I}$ and using the orthogonality of the base vectors, the following is obtained:

$$\dot{\bar{u}}_{iI} \Gamma_{\alpha I} - \dot{\bar{u}}_{i,j} x_{jI} \Gamma_{\alpha I} = \dot{u}_{iI} \gamma_{\alpha I} \tag{4.35}$$

With the definition of the mean velocity gradient, Equation 4.10, the nodal velocities above are eliminated. As a result, $\gamma_{\alpha I}$ is computed from the following expression:

$$\gamma_{\alpha I} = \Gamma_{\alpha I} - \frac{1}{V} B_{iI} x_{iJ} \Gamma_{\alpha J} \tag{4.36}$$

The difference between the hourglass base vectors $\Gamma_{\alpha I}$ and the hourglass shape vectors $\gamma_{\alpha I}$ is very important. They are identical if and only if the hexahedron is a right-parallelepiped. For a general shape, $\Gamma_{\alpha I}$ is orthogonal to B_{jI} while $\gamma_{\alpha I}$ is orthogonal to the linear velocity field $\dot{u}_{iI}^{\mathrm{lin}}$. While $\Gamma_{\alpha I}$ defines the hourglass pattern, $\gamma_{\alpha I}$ is necessary to accurately detect hourglassing.

For the purpose of controlling the hourglass modes, generalized forces $Q_{i\alpha}$ are defined conjugate to $\dot{q}_{i\alpha}$, so that the rate of work is

$$\dot{u}_{iI} f_{iI}^{\mathrm{hg}} = \frac{1}{2} Q_{i\alpha} \dot{q}_{i\alpha} \tag{4.37}$$

for arbitrary $\dot{u}_{iI}$. Using Equation 4.34, it follows that the contribution of the hourglass resistance to the nodal forces is given by

$$f_{iI}^{\mathrm{hg}} = \frac{1}{2} Q_{i\alpha} \gamma_{\alpha I} \ . \tag{4.38}$$

In JAC3D, an artificial stiffness resistance is used. In terms of the user-specifiable parameter κ, the resistance is given by

$$\dot{Q}_{i\alpha} = \frac{\kappa}{100} 2\hat{\mu} \frac{B_{jI} B_{jI}}{V} \dot{q}_{i\alpha} \ . \tag{4.39}$$

Note that the stiffness expression must be integrated, which requires that this resistance be stored in a global array. The term $\hat{\mu}$ is an effective shear modulus and is described below.

Observe that the nodal antihourglass forces of Equation 4.38 have the shape of $\gamma_{\alpha I}$ rather than $\Gamma_{\alpha I}$. This fact is essential, since the antihourglass forces should be orthogonal to the linear velocity field to prevent energy from being transferred to or from the rigid body and uniform strain modes by the antihourglassing scheme.

Determination of Effective Shear Modulus

The algorithm for calculating hourglass control requires an effective shear modulus. In JAC3D the following algorithm, similar to that found in PRONTO3D [7], is used for adaptively determining an effective shear modulus of the material.

The constitutive response over a time step can be cast as a hypoelastic relationship, and approximated as isotropic. This defines an effective shear modulus $\hat{\mu}$ in terms of the hypoelastic deviatoric stress and strain increments as follows:

$$s_{ij} = 2\hat{\mu} e_{ij} \Delta t \,, \tag{4.40}$$

where

$$s_{ij} = \Delta\sigma_{ij} - \frac{1}{3}\Delta\sigma_{kk}\delta_{ij} \tag{4.41}$$

and

$$e_{ij} = d_{ij} - \frac{1}{3} d_{kk}\delta_{ij} \,. \tag{4.42}$$

Taking the inner product of Equation 4.40 with the deviatoric strain rate and solving for the effective shear modulus $2\hat{\mu}$ gives

$$2\hat{\mu} = \frac{s_{ij} e_{ij}}{e_{mn} e_{mn} \Delta t} \,. \tag{4.43}$$

If the strain increments are insignificant, Equation 4.43 will not yield numerically meaningful results. In this circumstance, JAC3D sets the effective shear modulus to an initial estimate, μ_0. The effective modulus is not allowed to be greater than μ_0 or less than $\mu_0 \times 10^{-2}$.

4.1.2 Truss Element

To determine the contribution to nodal forces from the deformation of a truss element, the strain e_l is calculated as follows:

$$e_l = \frac{l_{n+1} - l_n}{l_{n+1/2}} \,, \tag{4.44}$$

where l is the length of the element. The subscript $n + 1/2$ denotes the length calculated midway through the load step. Once the stress in the truss has been determined from

e_l, the global components of force at the nodes are calculated by multiplying the stress times the truss cross-sectional area, and resolving the resulting axial force in the truss into global components based on the truss orientation. Changes in cross-sectional area due to elongation of the truss are not accounted for.

4.2 Finite Rotation Algorithm

As stated in Section 2.2, one of the fundamental numerical challenges in the development of an accurate algorithm for finite rotations was the determination of $\mathbf{R}$, the rotation tensor defined by the polar decomposition of the deformation gradient $\mathbf{F}$. An incremental algorithm is used for reasons of computational efficiency and numerical accuracy. This algorithm is identical to that used in PRONTO3D by Taylor and Flanagan [7]. The validity of the unrotated reference frame is based on the orthogonal transformation given by Equation 2.14. Therefore the crux of integrating Equation 2.6 for $\mathbf{R}$ is to maintain the orthogonality of $\mathbf{R}$. If one integrates $\dot{\mathbf{R}} = \mathbf{\Omega R}$ via a forward difference scheme, the orthogonality of $\mathbf{R}$ degenerates rapidly no matter how fine the time increments. The algorithm of Hughes and Winget [27] is adopted for integrating incremental rotations as follows.

A rigid body rotation over a time increment Δt may be represented by

$$\mathbf{x}_{t+\Delta t} = \mathbf{Q}_{\Delta t}\mathbf{x}_t \ , \tag{4.45}$$

where $\mathbf{Q}_{\Delta t}$ is a proper orthogonal tensor with the same rate of rotation as $\mathbf{R}$ given by Equation 2.6. The total rotation $\mathbf{R}$ is updated via the highly accurate expression below.

$$\mathbf{R}_{t+\Delta t} = \mathbf{Q}_{\Delta t}\mathbf{R}_t \tag{4.46}$$

For a constant rate of rotation, the midpoint velocity and the midpoint coordinates are related by

$$\frac{1}{\Delta t}(\mathbf{x}_{t+\Delta t} - \mathbf{x}_t) = \frac{1}{2}\mathbf{\Omega}(\mathbf{x}_{t+\Delta t} + \mathbf{x}_t) \ . \tag{4.47}$$

Combining Equations 4.45 and 4.47 yields

$$(\mathbf{Q}_{\Delta t} - \mathbf{I})\mathbf{x}_t = \frac{\Delta t}{2}\mathbf{\Omega}(\mathbf{Q}_{\Delta t} + \mathbf{I})\mathbf{x}_t \ . \tag{4.48}$$

Since $\mathbf{x}_t$ is arbitrary in Equation 4.48, it may be eliminated. Solving for $\mathbf{Q}_{\Delta t}$,

$$\mathbf{Q}_{\Delta t} = \left(\mathbf{I} - \frac{\Delta t}{2}\mathbf{\Omega}\right)^{-1}\left(\mathbf{I} + \frac{\Delta t}{2}\mathbf{\Omega}\right) \ . \tag{4.49}$$

The accuracy of this integration scheme is dependent upon the accuracy of the midpoint relationship of Equation 4.47. The rate of rotation must not vary significantly over

the time increment. Furthermore, Hughes and Winget [27] showed that the conditioning of Equation 4.49 degenerates as $\mathbf{\Omega}\Delta t$ grows.

The complete numerical algorithm for a single time step is shown in Table 4.6. This algorithm requires that the tensors $\mathbf{V}$ and $\mathbf{R}$ be stored in memory for each element.

Table 4.6. Finite Rotation Algorithm

1. Calculate	$\mathbf{D}$ and $\mathbf{W}$
2. Compute	$z_i = e_{ijk} V_{jm} D_{mk}$
	$\omega = \mathbf{w} - 2[\mathbf{V} - \mathbf{I}\,\mathrm{tr}\,\mathbf{V}]^{-1}\mathbf{z}$
	$\Omega_{ij} = \frac{1}{2} e_{ijk}\omega_k$
3. Solve	$(\mathbf{I} - \frac{\Delta t}{2}\mathbf{\Omega})\mathbf{R}_{t+\Delta t} = (\mathbf{I} + \frac{\Delta t}{2}\mathbf{\Omega})\mathbf{R}_t$
4. Calculate	$\dot{\mathbf{V}} = (\mathbf{D} + \mathbf{W})\mathbf{V} - \mathbf{V}\mathbf{\Omega}$
5. Update	$\mathbf{V}_{t+\Delta t} = \mathbf{V}_t + \Delta t \dot{\mathbf{V}}$
6. Compute	$\mathbf{d} = \mathbf{R}^T \mathbf{D} \mathbf{R}$
7. Integrate	$\dot{\sigma} = \mathbf{f}(\mathbf{d}, \sigma)$
8. Compute	$\mathbf{T} = \mathbf{R}\sigma\mathbf{R}^T$

5. Constitutive Models

The JAC3D program is written in modular form so that different material models can be added in the future. At the present time there are seven continuum material models and one truss model, although the isothermal elastic/plastic model is the only continuum model described here. Since each model is in some sense an independent module, each may be separately documented. Verification problems similar to those given in Chapter 8 should be included in the documentation. The input instructions given in Appendix A contain documentation references for the other models. Instructions for adding a new material model are given in Appendix C.

The function **f** in step 7 of Table 4.6 represents a stress-strain relationship. The main assumption is that the strain rate is constant from time t_{n-1} to t_n. During each conjugate gradient iteration the latest values of the kinematic quantities are used to update the stress. All material models are written in terms of the unrotated Cauchy stress $\boldsymbol{\sigma}$ and the deformation rate $\mathbf{d}$ in the unrotated configuration.

When calculating linear elastic material response, Hooke's law is used. In a rate form, this is written as

$$\dot{\boldsymbol{\sigma}} = \lambda(\operatorname{tr}\mathbf{d})\boldsymbol{\delta} + 2\mu\mathbf{d} \ , \tag{5.1}$$

where λ and μ are the elastic Lamé material constants.

5.1 Elastic/Plastic Material with Combined Hardening

The elastic/plastic model is based on a standard von Mises-type yield condition and uses combined kinematic and isotropic hardening. It behaves elastically if no yield stress is input. A very thorough description of this model is found in the PRONTO3D manual [7]. The description is repeated here for completeness.

5.1.1 Basic Definitions and Assumptions

Some definitions and assumptions are outlined here. Referring to Figure 5.1, which geometrically depicts the yield surface in deviatoric stress space, the backstress (the center of the yield surface) is defined by the tensor $\boldsymbol{\alpha}$. If $\boldsymbol{\sigma}$ is the current value of the stress, the deviatoric part of the current stress is

$$\mathbf{S} = \boldsymbol{\sigma} - \tfrac{1}{3}\operatorname{tr}\boldsymbol{\sigma}\boldsymbol{\delta} \ . \tag{5.2}$$

The stress difference is then measured by subtracting the backstress from the deviatoric stress by

$$\boldsymbol{\xi} = \mathbf{S} - \boldsymbol{\alpha} \ . \tag{5.3}$$

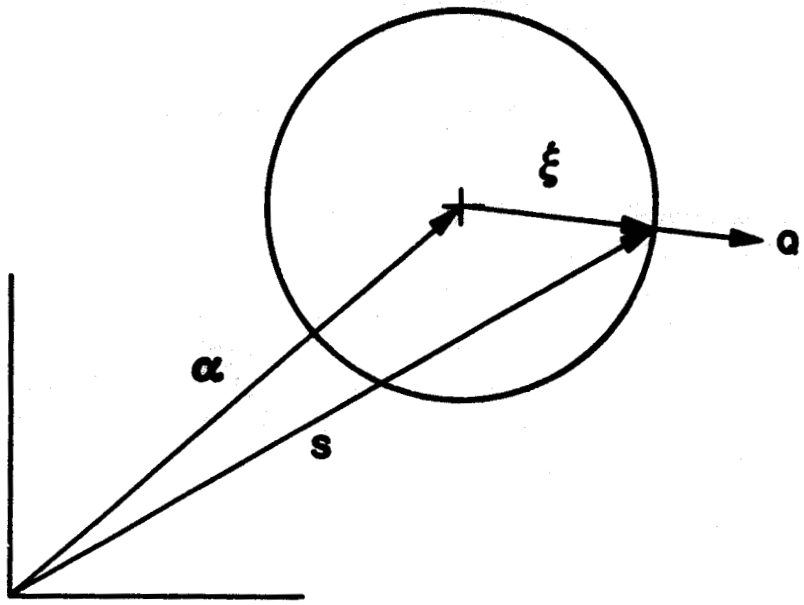


Figure 5.1. Yield Surface in Deviatoric Stress Space.

The magnitude of the deviatoric stress difference R is defined by

$$R = \|\boldsymbol{\xi}\| = \sqrt{\boldsymbol{\xi} : \boldsymbol{\xi}}\,, \tag{5.4}$$

where the inner product of second order tensors is $\mathbf{S} : \mathbf{S} = S_{ij}S_{ij}$. Note that if the backstress is zero (isotropic hardening case) the stress difference is equal to the deviatoric part of the current stress $\mathbf{S}$.

The von Mises yield surface is defined as

$$f(\boldsymbol{\sigma}) = \tfrac{1}{2}\boldsymbol{\xi} : \boldsymbol{\xi} = \kappa^2\,, \tag{5.5}$$

and the von Mises effective stress $\bar{\sigma}$ is defined by

$$\bar{\sigma} = \sqrt{\tfrac{3}{2}\boldsymbol{\xi} : \boldsymbol{\xi}}\ . \tag{5.6}$$

Since R is the magnitude of the deviatoric stress tensor when $\boldsymbol{\alpha} = 0$, it follows that

$$R = \sqrt{2}\kappa = \sqrt{\tfrac{2}{3}}\bar{\sigma}\ . \tag{5.7}$$

The normal to the yield surface can be determined from Equation 5.5

$$\mathbf{Q} = \frac{\partial f/\partial \boldsymbol{\sigma}}{\|\partial f/\partial \boldsymbol{\sigma}\|} = \frac{\boldsymbol{\xi}}{R} \tag{5.8}$$

It is assumed that the strain rate can be decomposed into elastic and plastic parts by an additive decomposition

$$\mathbf{d} = \mathbf{d}^{\mathrm{el}} + \mathbf{d}^{\mathrm{pl}} \tag{5.9}$$

and that the plastic part of the strain rate is given by a normality condition

$$\mathbf{d}^{\mathrm{pl}} = \gamma \mathbf{Q} \tag{5.10}$$

where the scalar multiplier γ is to be determined.

A scalar measure of equivalent plastic strain rate is defined by

$$\bar{d}^{\mathrm{pl}} = \sqrt{\tfrac{2}{3}\mathbf{d}^{\mathrm{pl}} : \mathbf{d}^{\mathrm{pl}}} \tag{5.11}$$

which is chosen such that

$$\bar{\sigma}\bar{d}^{\mathrm{pl}} = \boldsymbol{\sigma} : \mathbf{d}^{\mathrm{pl}} \ . \tag{5.12}$$

The stress rate is assumed to be purely due to the elastic part of the strain rate and is expressed in terms of Hooke's law by

$$\dot{\boldsymbol{\sigma}} = \lambda(\mathrm{tr}\,\mathbf{d}^{\mathrm{el}})\boldsymbol{\delta} + 2\mu\mathbf{d}^{\mathrm{el}} \tag{5.13}$$

where λ and μ are the Lamé constants for the material.

In what follows, the theory of isotropic hardening, kinematic hardening, and combined hardening is described.

5.1.2 Isotropic Hardening

In the isotropic hardening case, the backstress is zero and the stress difference is equal to the deviatoric stress $\mathbf{S}$. The consistency condition is written by taking the rate of Equation 5.5:

$$\dot{f}(\boldsymbol{\sigma}) = 2\kappa\dot{\kappa} \ . \tag{5.14}$$

The consistency condition requires that the state of stress must remain on the yield surface at all times. The chain rule and the definition of the normal to the yield surface given by Equation 5.8 is used to obtain

$$\dot{f}(\boldsymbol{\sigma}) = \frac{\partial f}{\partial \boldsymbol{\sigma}} : \dot{\boldsymbol{\sigma}} = \left\| \frac{\partial f}{\partial \boldsymbol{\sigma}} \right\| \mathbf{Q} : \dot{\boldsymbol{\sigma}} \tag{5.15}$$

and from Equations 5.4 and 5.5

$$\left\| \frac{\partial f}{\partial \boldsymbol{\sigma}} \right\| = \|\mathbf{S}\| = R \ . \tag{5.16}$$

Combining Equations 5.14, 5.15, and 5.16,

$$\frac{1}{R}\mathbf{S} : \dot{\boldsymbol{\sigma}} = \dot{R} \ . \tag{5.17}$$

Note that because $\mathbf{S}$ is deviatoric, $\mathbf{S} : \dot{\boldsymbol{\sigma}} = \mathbf{S} : \dot{\mathbf{S}}$ and

$$\mathbf{S} : \dot{\mathbf{S}} = \frac{d}{dt}\left(\tfrac{1}{2}\mathbf{S} : \mathbf{S}\right) = \frac{d}{dt}\left(\frac{\bar{\sigma}^2}{3}\right) = \tfrac{2}{3}\bar{\sigma}\dot{\bar{\sigma}} \ . \tag{5.18}$$

Then Equation 5.17 can be written as

$$\dot{R} = \sqrt{\tfrac{2}{3}}\dot{\bar{\sigma}} = \sqrt{\tfrac{2}{3}}H'\dot{\bar{d}}^{\mathrm{pl}} \tag{5.19}$$

where H' is the slope of the effective stress versus equivalent plastic strain ($\bar{\sigma}$ vs. $\bar{\epsilon}^{\mathrm{pl}}$). This may be derived from uniaxial tension test data as shown in Figure 5.2.

The consistency condition (Equation 5.17) and Equation 5.19 result in

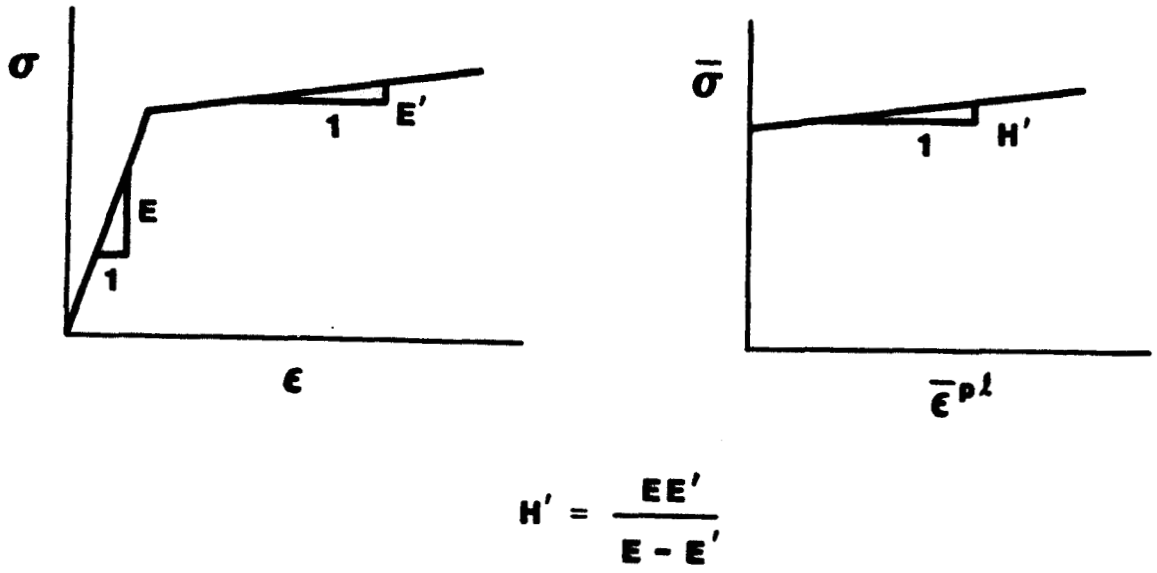
$$\sqrt{\tfrac{2}{3}}H'\dot{\bar{d}}^{\mathrm{pl}} = \mathbf{Q} : \dot{\boldsymbol{\sigma}} \ . \tag{5.20}$$

The trial elastic stress rate $\dot{\boldsymbol{\sigma}}^{\mathrm{tr}}$ is defined by

$$\dot{\boldsymbol{\sigma}}^{\mathrm{tr}} = \mathbf{C} : \mathbf{d} \tag{5.21}$$

where $\mathbf{C}$ is the fourth-order tensor of elastic coefficients defined by Equation 5.13. Combining the strain rate decomposition defined in Equation 5.9 with Equations 5.20 and 5.21 yields

$$\sqrt{\tfrac{2}{3}}H'\dot{\bar{d}}^{\mathrm{pl}} = \mathbf{Q} : \dot{\boldsymbol{\sigma}}^{\mathrm{tr}} - \mathbf{Q} : \mathbf{C} : \mathbf{d}^{\mathrm{pl}}. \tag{5.22}$$



$$H' = \frac{EE'}{E - E'}$$

Figure 5.2. Conversion of Data From a Uniaxial Tension Test to Equivalent Plastic Strain Versus von Mises Stress.

Since $\mathbf{Q}$ is deviatoric, $\mathbf{C}:\mathbf{Q} = 2\mu\mathbf{Q}$ and $\mathbf{Q}:\mathbf{C}:\mathbf{Q} = 2\mu$. Then using the normality condition (Equation 5.10), the definition of equivalent plastic strain (Equation 5.11), and Equation 5.22,

$$\tfrac{2}{3}H'\gamma = \mathbf{Q}:\dot{\boldsymbol{\sigma}}^{\mathrm{tr}} - 2\mu\gamma \tag{5.23}$$

and since $\mathbf{Q}$ is deviatoric ($\mathbf{Q}:\dot{\boldsymbol{\sigma}}^{\mathrm{tr}} = 2\mu\mathbf{Q}:\mathbf{d}$), γ is determined from Equation 5.23 as

$$\gamma = \frac{1}{(1+\frac{H'}{3\mu})}\mathbf{Q}:\mathbf{d} \ . \tag{5.24}$$

The current normal to the yield surface $\mathbf{Q}$ and the total strain rate $\mathbf{d}$ are known quantities. Hence, from Equation 5.24, γ can be determined and then used in Equation 5.10 to calculate the plastic part of the strain rate. With the additive strain rate decomposition and the elastic stress rate of Equations 5.9 and 5.13, this completes the definition of the rate equations.

The means of integrating the rate equations, subject to the constraint that the stress must remain on the yield surface, still remains to be explained. How that is accomplished will be shown in Section 5.1.5.

5.1.3 Kinematic Hardening

For kinematic hardening, the von Mises yield condition is written in terms of the stress difference $\boldsymbol{\xi}$:

$$f(\boldsymbol{\xi}) = \tfrac{1}{2}\boldsymbol{\xi}:\boldsymbol{\xi} = \kappa^2 \ . \tag{5.25}$$

It is important to remember that both $\boldsymbol{\xi}$ and the back stress $\boldsymbol{\alpha}$ are deviatoric tensors. The consistency condition for kinematic hardening is written as

$$\dot{f}(\boldsymbol{\xi}) = 0 \tag{5.26}$$

because the size of the yield surface does not grow with kinematic hardening ($\dot{\kappa} = 0$). Using the chain rule on Equation 5.26,

$$\frac{\partial f}{\partial \boldsymbol{\xi}}:\dot{\boldsymbol{\xi}} = 0 \tag{5.27}$$

and

$$\frac{\partial f}{\partial \boldsymbol{\xi}} = \left\|\frac{\partial f}{\partial \boldsymbol{\xi}}\right\|\mathbf{Q} = R\mathbf{Q} \ . \tag{5.28}$$

Combining Equations 5.27 and 5.28 and assuming that $R \neq 0$,

$$\mathbf{Q}:\dot{\boldsymbol{\xi}} = 0 \tag{5.29}$$

or

$$\mathbf{Q}:(\dot{\mathbf{S}} - \dot{\boldsymbol{\alpha}}) = 0 \ . \tag{5.30}$$

A geometric interpretation of Equation 5.30 is shown in Figure 5.3 where it can be seen that the backstress moves in a direction parallel to the normal to the yield surface.

The back stress rate $\dot{\boldsymbol{\alpha}}$ must now be defined. Recall that for the isotropic hardening case (Equation 5.20),

$$\mathbf{Q} : \dot{\boldsymbol{\sigma}} = \sqrt{\tfrac{2}{3}} H' \dot{\bar{d}}^{\mathrm{pl}} = \tfrac{2}{3} H' \gamma \ . \tag{5.31}$$

The kinematic hardening condition assumes that

$$\dot{\boldsymbol{\alpha}} = \phi \mathbf{d}^{\mathrm{pl}} = \phi \gamma \mathbf{Q} \ , \tag{5.32}$$

where ϕ is a material parameter. If ϕ is chosen to be $\frac{2}{3}H'$, Equations 5.32 and 5.30 give a result identical to the isotropic hardening case (Equation 5.31). Hence, either Equation 5.31 or 5.32 gives us a scalar condition on $\dot{\boldsymbol{\alpha}}$. Note that both of these are assumptions and must be shown to be reasonable. Experience with material models based on these assumptions has shown that, in fact, they are reasonable representations of material behavior.

Using Equation 5.32, Equation 5.9 (the strain rate decomposition), and Equation 5.13 (the elastic stress rate) in Equation 5.30 (the consistency condition for kinematic hardening) gives

$$\mathbf{Q} : (\dot{\boldsymbol{\sigma}}^{\mathrm{tr}} - \mathbf{C} : \mathbf{d}^{\mathrm{pl}}) = \mathbf{Q} : \tfrac{2}{3} H' \gamma \mathbf{Q} \tag{5.33}$$

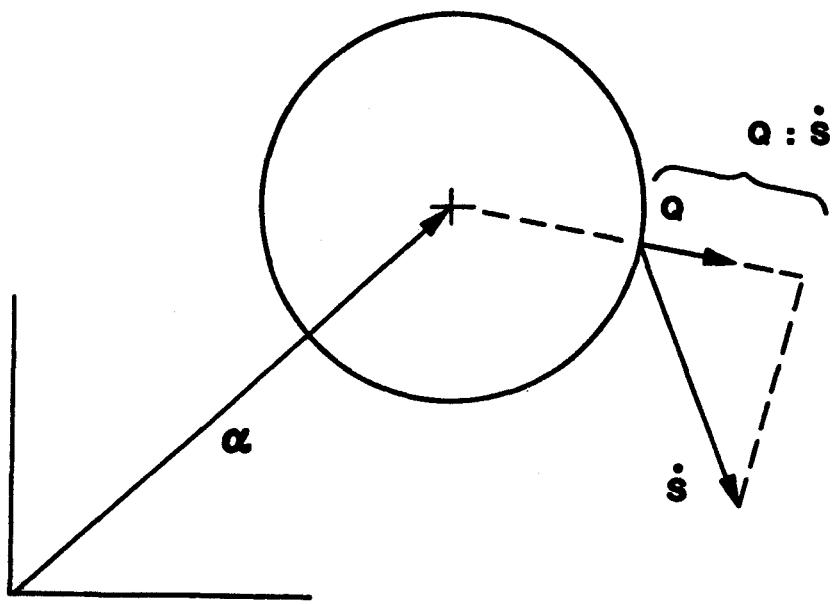


Figure 5.3. Geometric Interpretation of the Consistency Condition for Kinematic Hardening.

After using the normality condition ($\mathbf{d}^{pl} = \gamma\mathbf{Q}$), we note that because $\mathbf{Q}$ is deviatoric, $\mathbf{C} : \mathbf{Q} = 2\mu\mathbf{Q}$. Solving Equation 5.33 for γ then gives

$$\gamma = \frac{1}{(1 + \frac{H'}{3\mu})}\mathbf{Q} : \mathbf{d} \tag{5.34}$$

which is the same result as was obtained for the isotropic hardening case.

5.1.4 Combined Isotropic and Kinematic Hardening

For the combined hardening case we define a scalar parameter β, ranging from 0 to 1, which determines the relative amount of each type of hardening. Figure 5.4 illustrates the uniaxial response to reversed loading which results from different choices of β. When $\beta = 0$, only kinematic hardening occurs and when $\beta = 1$, only isotropic hardening occurs.

The results derived for the independent hardening cases are multipled by the appropriate fraction for each type of hardening. Equations 5.19 and 5.32 are rewritten as

$$\dot{R} = \sqrt{\tfrac{2}{3}}H'\dot{\bar{d}}^{pl}\beta \tag{5.35}$$

and

$$\dot{\boldsymbol{\alpha}} = \tfrac{2}{3}H'\mathbf{d}^{pl}(1-\beta) = \tfrac{2}{3}H'\gamma\mathbf{Q}(1-\beta) \ . \tag{5.36}$$

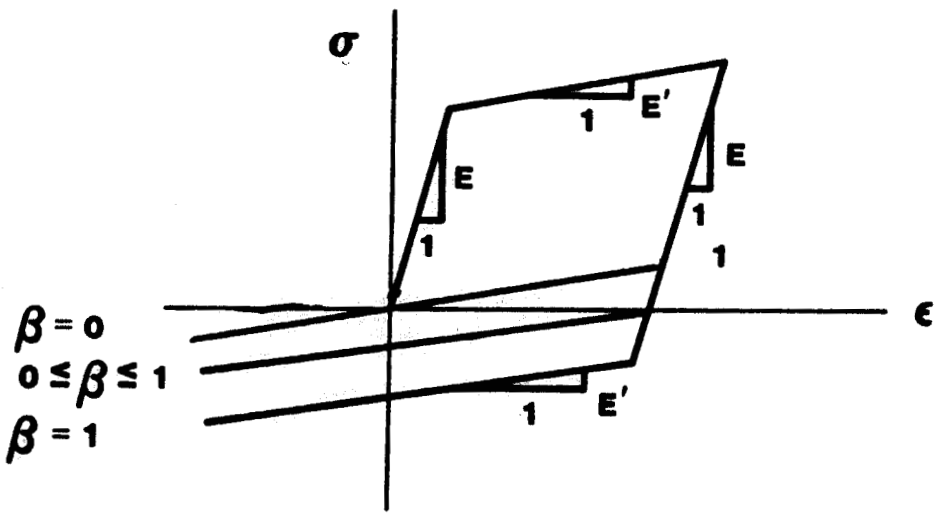


Figure 5.4. Effect of the Hardening Parameter β on Uniaxial Response.

As before, the consistency condition is

$$\mathbf{Q}:\dot{\boldsymbol{\xi}} = \dot{R} \tag{5.37}$$

or

$$\mathbf{Q}:(\dot{\mathbf{S}} - \dot{\boldsymbol{\alpha}}) = \sqrt{\tfrac{2}{3}}H'\dot{\bar{d}}^{\mathrm{pl}}\beta \ . \tag{5.38}$$

Recall, as before, that $\mathbf{Q}:\dot{\mathbf{S}} = \mathbf{Q}:(\dot{\boldsymbol{\sigma}}^{\mathrm{tr}} - \gamma\mathbf{C}:\mathbf{Q})$, using the elastic stress rate, the additive strain rate decomposition, and the normality condition. Together with Equations 5.36 and 5.11, this transforms Equation 5.38 into

$$\mathbf{Q}:\dot{\boldsymbol{\sigma}}^{\mathrm{tr}} - \gamma\mathbf{Q}:\mathbf{C}:\mathbf{Q} - \tfrac{2}{3}H'\gamma(1-\beta)\mathbf{Q}:\mathbf{Q} = \sqrt{\tfrac{2}{3}}H'\beta\sqrt{\tfrac{2}{3}(\gamma\mathbf{Q}):(\gamma\mathbf{Q})} \ . \tag{5.39}$$

Solving for γ,

$$\gamma = \frac{1}{(1+\frac{H'}{3\mu})}\mathbf{Q}:\mathbf{d} \tag{5.40}$$

which is the same result as was obtained for each of the independent cases.

The following is a summary of the governing equations for the combined theory:

$$\dot{\boldsymbol{\sigma}} = \mathbf{C}:(\mathbf{d} - \mathbf{d}^{\mathrm{pl}}) = \dot{\boldsymbol{\sigma}}^{\mathrm{tr}} - 2\mu\gamma\mathbf{Q} \tag{5.41}$$

$$\dot{R} = \beta\sqrt{\tfrac{2}{3}}H'\dot{\bar{d}}^{\mathrm{pl}} = \beta\tfrac{2}{3}H'\gamma \tag{5.42}$$

$$\dot{\boldsymbol{\alpha}} = (1-\beta)\tfrac{2}{3}H'\mathbf{d}^{\mathrm{pl}} \tag{5.43}$$

$$\mathbf{d}^{\mathrm{pl}} = \begin{cases} 0 \text{ (elastic), if } f(\boldsymbol{\xi}) < \kappa^2 \\ \gamma\mathbf{Q} \text{ (plastic), if } f(\boldsymbol{\xi}) \geq \kappa^2 \end{cases} \tag{5.44}$$

$$\gamma = \frac{1}{(1+\frac{H'}{3\mu})}\mathbf{Q}:\mathbf{d} \tag{5.45}$$

$$\mathbf{Q} = \frac{\partial f/\partial\boldsymbol{\sigma}}{\|\partial f/\partial\boldsymbol{\sigma}\|} = \frac{\boldsymbol{\xi}}{R} \tag{5.46}$$

5.1.5 Numerical Implementation

The finite element algorithm requires an incremental form of Equations 5.41 through 5.46. Additionally, an algorithm must be used that integrates the incremental equations subject to the constraint that the stress remains on the yield surface.

The incremental analogs of Equations 5.41 through 5.43 are

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}^{\mathrm{tr}}_{n+1} - 2\mu\Delta\gamma\mathbf{Q} \tag{5.47}$$

$$R_{n+1} = R_n + \tfrac{2}{3}\beta H'\Delta\gamma \tag{5.48}$$

and

$$\boldsymbol{\alpha}_{n+1} = \boldsymbol{\alpha}_n + (1-\beta)\tfrac{2}{3}H'\Delta\gamma\mathbf{Q} \tag{5.49}$$

where $\Delta\gamma$ represents the product of the time increment and the equivalent plastic strain rate ($\Delta\gamma = \gamma\Delta t$). The subscripts n and $n+1$ refer to the beginning and end of a time step, respectively.

An incremental analog is needed for the rate forms of the consistency condition given by Equations 5.14, 5.26, and 5.38. At the end of the time step, the stress state must be on the yield surface. Hence, the incremental consistency condition is

$$\boldsymbol{\alpha}_{n+1} + R_{n+1}\mathbf{Q} = \mathbf{S}_{n+1} \ . \tag{5.50}$$

Equation 5.50 is depicted graphically in Figure 5.5.

Substituting the definitions given by Equations 5.47 through 5.49 into the consistency condition of Equation 5.50,

$$\left[\boldsymbol{\alpha}_n + (1-\beta)\tfrac{2}{3}H'\Delta\gamma\mathbf{Q}\right] + \left[R_n + \tfrac{2}{3}\beta H'\Delta\gamma\right]\mathbf{Q} = \mathbf{S}^{\mathrm{tr}}_{n+1} - 2\mu\Delta\gamma\mathbf{Q} \ . \tag{5.51}$$

Taking the tensor product of both sides of Equation 5.51 with $\mathbf{Q}$ and solving for $\Delta\gamma$,

$$\Delta\gamma = \frac{1}{2\mu}\frac{1}{(1+\frac{H'}{3\mu})}(\|\boldsymbol{\xi}^{\mathrm{tr}}_{n+1}\| - R_n) \ . \tag{5.52}$$

It follows from Equation 5.52 that the plastic strain increment is proportional to the magnitude of the excursion of the elastic trial stress past the yield surface (see Figure 5.6).

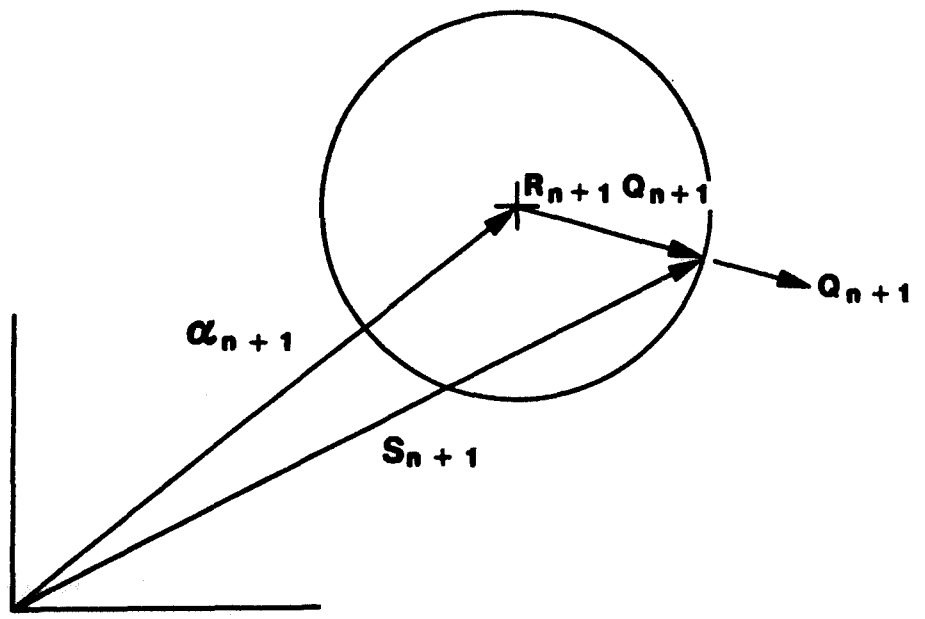


Figure 5.5. Geometric Interpretation of the Incremental Form of the Consistency Condition for Combined Hardening.

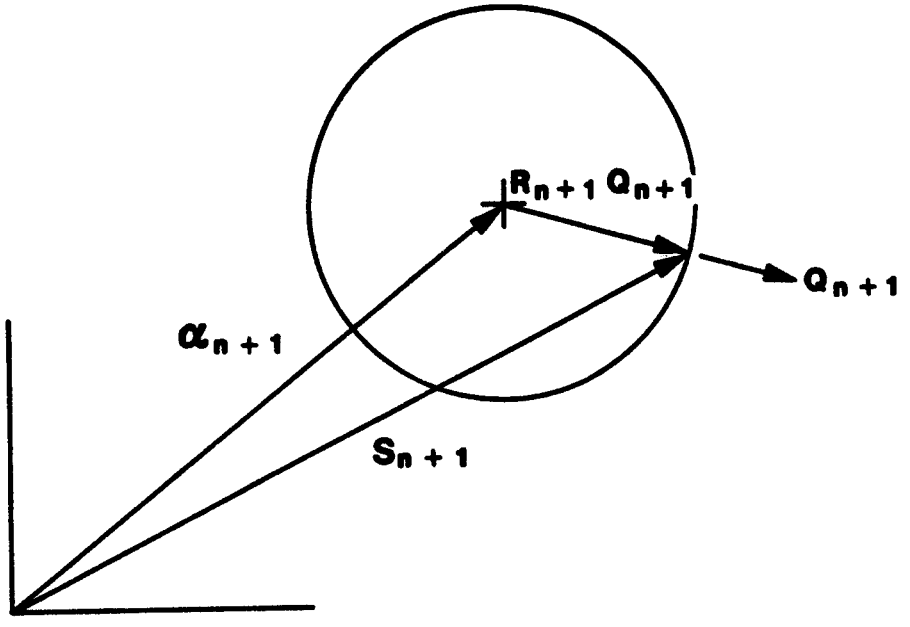


Figure 5.6. Geometric Interpretation of the Radial Return Correction.

Using the result of Equation 5.52 in Equations 5.47 through 5.49 completes the algorithm. In addition,

$$\Delta \mathbf{d}^{\mathrm{pl}} = \Delta\gamma \mathbf{Q} \tag{5.53}$$

and

$$\Delta \bar{d}^{\mathrm{pl}} = \sqrt{\tfrac{2}{3}}\Delta\gamma \ . \tag{5.54}$$

Using Equation 5.52 in Equation 5.47 shows that the final stress is calculated by returning the elastic trial stress radially to the yield surface at the end of the time step (hence the name Radial Return Method). Estimates of the accuracy of this method and other methods for similarly integrating the rate equations are available in Krieg and Krieg [28] and Schreyer, et al. [29]. Note that the radial return correction (the last term in Equation 5.47) is purely deviatoric.

5.1.6 Secant Modulus

A secant modulus is needed to make the conjugate gradient solution algorithm more efficient. To derive the secant modulus, Equation 5.21 is written as

$$\sigma^{\mathrm{tr}}_{n+1} = \sigma_n + \mathbf{C} : \mathbf{d}\Delta t \tag{5.55}$$

for calculating the trial stress state with the elastic modulus $\mathbf{C}$. Next, Equation 5.47 is written as

$$\boldsymbol{\sigma}_{n+1} = \boldsymbol{\sigma}^{\mathrm{tr}}_{n+1} - a\boldsymbol{\xi}^{\mathrm{tr}}_{n+1} \tag{5.56}$$

where

$$a = b\left(1 - \frac{R_n}{\sqrt{\boldsymbol{\xi}^{\mathrm{tr}}_{n+1} : \boldsymbol{\xi}^{\mathrm{tr}}_{n+1}}}\right) \quad . \tag{5.57}$$

and

$$b = \frac{1}{(1 + \frac{H'}{3\mu})} \tag{5.58}$$

By using the operator $\mathbf{D}$ defined by

$$\mathbf{D} = \begin{bmatrix} +2/3 & -1/3 & -1/3 & 0 & 0 & 0 \\ -1/3 & +2/3 & -1/3 & 0 & 0 & 0 \\ -1/3 & -1/3 & +2/3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} , \tag{5.59}$$

Equation 5.55 can be rewitten as

$$\boldsymbol{\xi}^{\mathrm{tr}}_{n+1} = \boldsymbol{\xi}_n + \mathbf{D} : \mathbf{C} : \mathbf{d}\Delta t \tag{5.60}$$

Now Equations 5.55, 5.56, and 5.60 are combined to eliminate all trial values and obtain

$$\boldsymbol{\sigma}_{n+1} - \boldsymbol{\sigma}_n = (\mathbf{I} - a\mathbf{D}) : \mathbf{C} : \mathbf{d}\Delta t - a\,\boldsymbol{\xi}_n \quad . \tag{5.61}$$

With the use of Equation 5.60, the expression $\left(\boldsymbol{\xi}^{\mathrm{tr}}_{n+1} : \boldsymbol{\xi}^{\mathrm{tr}}_{n+1}\right)^{-1/2}$ in Equation 5.57 is approximated with a two-term expansion as

$$\left(\boldsymbol{\xi}^{\mathrm{tr}}_{n+1} : \boldsymbol{\xi}^{\mathrm{tr}}_{n+1}\right)^{-1/2} \cong \frac{1}{\sqrt{\boldsymbol{\xi}^{\mathrm{tr}}_{n+1} : \boldsymbol{\xi}_{n+1}}}\left(1 - \frac{\boldsymbol{\xi}^{\mathrm{tr}}_{n+1} : \mathbf{C} : \mathbf{d}\Delta t}{2\,\boldsymbol{\xi}^{\mathrm{tr}}_{n+1} : \boldsymbol{\xi}_{n+1}}\right) \tag{5.62}$$

Substituting Equation 5.62 into Equation 5.61 results in

$$\boldsymbol{\sigma}_{n+1} - \boldsymbol{\sigma}_n = \left(\mathbf{I} - a\mathbf{D} - \frac{bR_n\boldsymbol{\xi}_{n+1} \otimes \boldsymbol{\xi}^{\mathrm{tr}}_{n+1}}{2(\boldsymbol{\xi}^{\mathrm{tr}}_{n+1} : \boldsymbol{\xi}_{n+1})^{3/2}}\right) : \mathbf{C} : \mathbf{d}\Delta t - b\left(1 - \frac{R_n}{\sqrt{\boldsymbol{\xi}^{\mathrm{tr}}_{n+1} : \boldsymbol{\xi}_{n+1}}}\right)\boldsymbol{\xi}_n \tag{5.63}$$

Seeking a secant modulus $\mathbf{C}^*$ such that

$$\boldsymbol{\sigma}_{n+1} - \boldsymbol{\sigma}_n = \mathbf{C}^* : \mathbf{d}\Delta t \quad . \tag{5.64}$$

we drop the last term in Equation 5.63 and approximate $\mathbf{C}^*$ by replacing $\boldsymbol{\xi}_{n+1}$ with $\boldsymbol{\xi}^{\mathrm{tr}}_{n+1}$:

$$\mathbf{C}^* \cong \left(\mathbf{I} - a\mathbf{D} - \frac{bR_n\boldsymbol{\xi}^{\mathrm{tr}}_{n+1} \otimes \boldsymbol{\xi}^{\mathrm{tr}}_{n+1}}{2\|\boldsymbol{\xi}^{\mathrm{tr}}_{n+1}\|^3}\right) : \mathbf{C} \quad . \tag{5.65}$$

5.2 Elastic Truss Material

The incremental stress in an elastic truss is computed by multiplying the strain increment e_l (Equation 4.44) by Young's modulus. The new stress, then, is given by

$$\sigma_{n+1} = \sigma_n + Ee_l \quad . \tag{5.66}$$

6. Contact Surfaces

Many structures consist of two or more parts that are in contact and slide with respect to one another. In the setting of the CG method, a sliding algorithm can easily be incorporated using the master-slave concept. When interference between two surfaces is detected, the nodes on the slave surface are constrained to move on the master surface. Friction can be used to constrain differential motion of the two surfaces in the direction tangent to the master surface. The method is also conveniently used to fix two surfaces together. The master-slave concept is also discussed by Stone et. al. [30].

As mentioned above, the master-slave algorithm will keep nodes on the slave surface from penetrating the master surface. Thus, if a slave node moves past the end of a master surface, the desired constraint may not be present. This problem can usually be fixed by reversing the master and slave surface designations. Also, in order to obtain the highest degree of constraint, it is best to have the more coarsely-meshed surface be the master surface.

6.1 Search Algorithm

The relationship between a slave node and its master surface is shown in Figure 6.1.

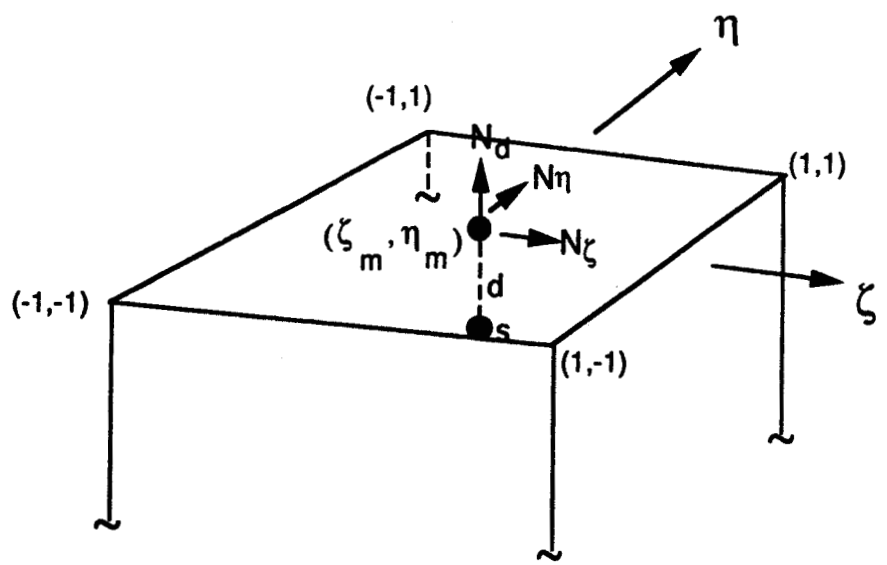


Figure 6.1. Master-Slave Relationships for Sliding Interfaces.

The slave node S has penetrated the master surface element face and is located at a perpendicular distance d from the master surface. A local coordinate system using isoparametric coordinates ζ and η is used to describe the master surface by

$$x = f(\zeta, \eta) \tag{6.1}$$

In order to find the location point M on the master surface, (ζ_M, η_M) must be found such that the vector d is perpendicular to N_ζ and N_η. To proceed, the functions Φ_1 and Φ_2 are defined as follows ("$\cdot$" represents the dot product of two vectors):

$$\Phi_1 = d \cdot N_\zeta \tag{6.2}$$

$$\Phi_2 = d \cdot N_\eta \ . \tag{6.3}$$

The quantities d, N_ζ, and N_η are

$$d = f - x_S \tag{6.4}$$

$$N_\zeta = \frac{\partial f}{\partial \zeta} \tag{6.5}$$

$$N_\eta = \frac{\partial f}{\partial \eta} \tag{6.6}$$

where x_S is the location of the slave node. Approaching the desired point (ζ_M, η_M) will drive Φ_1 and Φ_2 to zero.

Following Benson [31], the calculation of the location of the slave node on the master surface is begun with three iterations of a least-squares projection used to generate a starting guess.

$$\zeta_0 = 0 \quad \eta_0 = 0 \tag{6.7}$$

$$\begin{bmatrix} N_\zeta \\ N_\eta \end{bmatrix} [N_\zeta N_\eta] \begin{bmatrix} \Delta\zeta \\ \Delta\eta \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix} \tag{6.8}$$

$$\zeta_k = \zeta_{k-1} + \Delta\zeta \tag{6.9}$$

$$\eta_k = \eta_{k-1} + \Delta\eta \tag{6.10}$$

Then a Newton-Raphson procedure is used with the following equations to solve for ζ_M and η_M.

$$\begin{bmatrix} \frac{\partial \Phi_1}{\partial \zeta} & \frac{\partial \Phi_1}{\partial \eta} \\ \frac{\partial \Phi_2}{\partial \zeta} & \frac{\partial \Phi_2}{\partial \eta} \end{bmatrix} \begin{bmatrix} \Delta\zeta \\ \Delta\eta \end{bmatrix} = - \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix} \tag{6.11}$$

$$\zeta_k = \zeta_{k-1} + \Delta\zeta \tag{6.12}$$

$$\eta_k = \eta_{k-1} + \Delta\eta \tag{6.13}$$

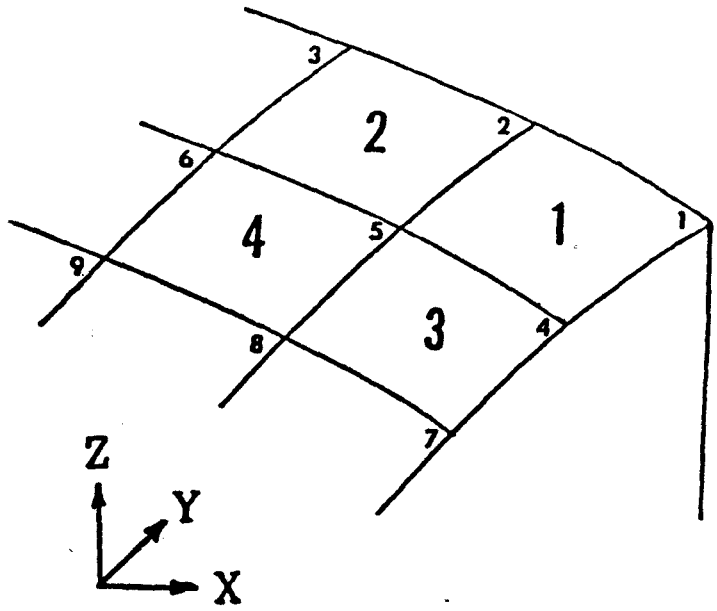
After the first global CG iteration, the procedure is accelerated by storing (ζ_M, η_M) for use as an initial guess for the next CG iteration. Convergence is assumed if $\Delta\zeta_k$ and $\Delta\eta_k$ are less than 1×10^{-5}.

To aid in searching for the element face that has the greatest possibility of being the surface which is penetrated by the slave node, a data base for each master face is used. For each master element face, a list of neighboring master surface faces is stored in an array. If the result of the calculation for the slave node location results in an absolute value of ζ_M or η_M greater than 1.0001, then the neighbor in the direction indicated by the signs of ζ_M and η_M is used in the next search calculation. An example is shown in Figure 6.2, where the neighboring face array **IFACE** for master face 1 is $(2, 3, 0, 0)$.

If d is less than a user-specified fraction (*stolr*) of the master-face length, the normal component of the slave residual force R_{Sn} is less than the user-specified tensile capacity (*ftolr*) of the interface, and

$$-1.0001 < \zeta_M, \eta_M < 1.0001 \,, \tag{6.14}$$

it is deemed that the slave node has penetrated the master face.



IFACE(1) = [2,3,0,0]

Figure 6.2. Example of the Neighboring Face Array.

6.2 Kinematic and Force Conditions

The kinematic and force conditions that apply to a contact surface are very similar to those in a finite-element assembly process. The conditions are applied to both the residual force vector R_j and the conjugate gradient vector P_j.

6.2.1 Force Conditions

Fixed Interface

For a fixed interface, all the components of the slave node residual force vector are applied as point loads to the master face at the location (ζ_M, η_M). The result is a set of forces that are distributed to the nodes associated with the master face by

$$R_{Mi\alpha} = \phi_\alpha R_{Si} \tag{6.15}$$

where the ϕ_α are interpolation functions of the master face. Then all the residual force components at the slave node are set to zero. This action ensures that the total force remains constant for the problem, and that the norms associated with the CG method are correctly calculated. If the slave node is directly in contact with a master node, this process is exactly the same as a finite-element assembly process. In addition, all slave nodes in contact with a master surface are subjected to linear constraint conditions defined by their location on the master surface.

Sliding Interface

For a sliding interface, simple Coulomb friction conditions have been implemented in the code. The same actions are applied as in the case of a fixed interface with respect to the force transfer from the slave node to the master surface. However, the maximum amount of force transferred in the direction tangent to the master surface is limited to the value of the friction coefficient times the normal slave force. The tangent direction is calculated by projecting the velocity of the slave node onto the master surface.

6.2.2 Kinematic Conditions

To ensure that the slave node moves properly on the master surface, kinematic conditions are applied to the solution process by modification of the slave node CG vector P_S before the line search is undertaken. Because updates to the velocity vector of the slave node are linearly dependent upon P_S, any modification to P_S will be reflected in the velocity vector update after the CG line search is performed.

First, the P-vectors at the four master-surface nodes are interpolated to the location of the slave node to define P_M, the motion of the master surface. P_M is then rotated to the (n, ζ, η) coordinate system. A $\hat{P}$ for the slave node is then constructed as follows.

For the normal component,

$$\hat{P}_n = P_{Mn} - \frac{0.2d}{\alpha_j} \tag{6.16}$$

where P_{Mn} is the component of P_M in the surface normal direction. The subtraction of $0.2d/\alpha_j$ will result in slowly pushing the slave node back to the master surface every CG iteration. In practice, a factor of one is used in place of α_j for the following reasons. First, α_j is not known until after the line search is performed. Second, α_j is usually ends up close to one due to the choice of the CG preconditioning matrix M. Third, the factor 0.2 is somewhat arbitrary anyway.

If a fixed interface is specified, then the transverse components of $\hat{P}$ are

$$\begin{aligned} \hat{P}_\zeta &= P_{M\zeta} \\ \hat{P}_\eta &= P_{M\eta} \end{aligned} \tag{6.17}$$

where $P_{M\zeta}$ and $P_{M\eta}$ are components of P_M in the ζ- and η-directions, respectively. Otherwise, for a sliding interface, the slave node retains its original transverse components:

$$\begin{aligned} \hat{P}_\zeta &= P_{S\zeta} \\ \hat{P}_\eta &= P_{S\eta} \end{aligned} \tag{6.18}$$

After $\hat{P}$ is constructed, it is rotated back to the global coordinate system and substituted for the original slave node CG vector.

6.3 Diagonal Assembly

For all of the sliding conditions, the preconditioning matrix (which is simply the diagonal of the linear stiffness matrix) must be assembled correctly to account for the fact that two surfaces are in contact. The diagonal term associated with the slave node is distributed to the master-face nodes using the interpolation functions ϕ_α:

$$M_{M\alpha} = \phi_\alpha M_S \tag{6.19}$$

The distributed slave values are then added to the diagonal values for the master nodes. Again, as with the residual force assembly, this action ensures that the generalized CG vector Z_j is properly calculated.

Blank Page

7. Loads and Boundary Conditions

JAC3D supports several types of loads and boundary conditions. Displacements, pressures, concentrated forces, and body forces may be prescribed. This chapter describes how these are implemented in the program.

7.1 Kinematic Boundary Conditions

The kinematic boundary conditions described below are *all* accomplished by altering the residual vector during the CG iterative process at the nodal points. All of the kinematic boundary conditions are applied to nodal point sets.

7.1.1 Zero-Displacement Constraints

A zero-displacement constraint is accomplished by setting the appropriate component of residual force at each selected node to zero during the CG iterative process. This will kinematically constrain the problem because the update to the velocity vector using Equations 3.11 and 3.13 is linearly dependent upon R_j. The starting value for the appropriate displacement component must also be initialized to zero. Zero-displacement constraints may be specified in any of the coordinate directions, or normal to any plane parallel to the z-axis.

7.1.2 Nonzero-Displacement Constraints

A nonzero-displacement constraint is specified by initializing the component of velocity with the change in displacement needed to satisfy the specification. The starting value for the appropriate displacement component must also be initialized to the value specified for the end of the time step. Then setting to zero the same component of the residual force vector to zero for all conjugate gradient iterations will ensure that the initial values will not change with each conjugate gradient iteration. This action will result in $\dot{u}_i^{j+1} = \dot{u}_i^j$. Nonzero-displacement constraints may be specified in any of the coordinate directions.

7.1.3 General Skewed-Displacement Constraints

Displacement constraints in an arbitrary direction are not explicitly programmed in JAC3D. However, this type of constraint is easily imposed with the use of a sliding interface. First, elements are defined that describe the surface on which the body will slide. The surface nodes of these elements are then given zero- or nonzero-displacement

constraints to reflect the desired rigid body motion required of the surface. The master element block must be given material properties, even though it will behave as a rigid body. The surface of the body to be constrained is defined as the slave side of a sliding interface, while the rigid surface is defined to be its master side as described in Chapter 6.

7.2 Traction Boundary Conditions and Distributed Loads

The boundary conditions described here apply external forces to selected nodes. The pressure boundary condition deals with element side sets, while the nodal force boundary condition applies to nodal point sets. Element side sets and nodal point sets are discussed in the EXODUS manual [19]. Body forces (distributed loads) are applied to each node in proportion to the mass of the material that surrounds it.

7.2.1 Pressure

The set of consistent nodal point forces arising from pressures distributed over an element side are defined by the last integral in Equation 3.6 by

$$\delta\dot{u}_{iI} f_{iI} = \delta\dot{u}_{iI} \int_S \phi_I(-p n_i)\, dA \tag{7.1}$$

where the range of the lowercase subscripts (coordinate directions) is 3, while the range of uppercase subscripts (surface nodes) is 4. Since the virtual velocities are arbitrary, they may be eliminated to yield

$$f_{iI} = -\int_S \phi_I p n_i\, dA \ . \tag{7.2}$$

The most general pressure distribution allowed is a mapping from nodal point pressure values via the isoparametric shape functions. The resulting expression for the consistent nodal forces is

$$f_{iI} = -p_J \int_S \phi_I \phi_J n_i\, dA \ . \tag{7.3}$$

For the surface of the eight-node uniform stress element used in JAC3D, ϕ_I is given by

$$\phi_I = \tfrac{1}{4}\Sigma_I + \tfrac{1}{2}\xi\Lambda_{1I} + \tfrac{1}{2}\eta\Lambda_{2I} + \xi\eta\Gamma_I \tag{7.4}$$

where

$$\Sigma_I = \begin{Bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{Bmatrix} \quad \Lambda_{1I} = \begin{Bmatrix} -1 \\ 1 \\ 1 \\ -1 \end{Bmatrix} \quad \Lambda_{2I} = \begin{Bmatrix} -1 \\ -1 \\ 1 \\ 1 \end{Bmatrix} \quad \Gamma_I = \begin{Bmatrix} 1 \\ -1 \\ 1 \\ -1 \end{Bmatrix} \tag{7.5}$$

and $n_i n_i = 1$. Figure 7.1 shows geometric definitions of a pressure loading.

It follows that

$$x_i = x_{iI}\phi_I \tag{7.6}$$

and

$$\begin{aligned} n_i\,dA &= e_{ijk}\frac{\partial x_J}{\partial \xi}\frac{\partial x_k}{\partial \eta}\,d\xi\,d\eta \\ &= e_{ijk}\,x_{jM}\,x_{kN}(\tfrac{1}{2}\Lambda_{1M} + \eta\Gamma_M)(\tfrac{1}{2}\Lambda_{2N} + \xi\Gamma_N)\,d\xi\,d\eta \ . \end{aligned} \tag{7.7}$$

Then the consistent nodal forces can be written as

$$f_{iI} = -p_J\,e_{ijk}\,x_{jM}\,x_{kN}\int_{-\frac{1}{2}}^{\frac{1}{2}}\int_{-\frac{1}{2}}^{\frac{1}{2}}\phi_I\phi_J(\tfrac{1}{2}\Lambda_{1M} + \eta\Gamma_M)(\tfrac{1}{2}\Lambda_{2N} + \xi\Gamma_N)\,d\xi\,d\eta \ . \tag{7.8}$$

The above integral involves 64 terms. Only terms with even powers of both ξ and η are nonzero. Of the remaining 16 terms, 4 vanish due to the properties of the alternator. The final 12 terms are given below.

$$\begin{aligned} f_{iI} = -p_J\,e_{ijk}\,x_{jM}\,x_{kN}\int_{-\frac{1}{2}}^{\frac{1}{2}}\int_{-\frac{1}{2}}^{\frac{1}{2}}&(\tfrac{1}{2}\Sigma_I + \tfrac{1}{2}\xi\Lambda_{1I} + \tfrac{1}{2}\eta\Lambda_{2I} + \xi\eta\Gamma_I) \\ &(\tfrac{1}{2}\Sigma_J + \tfrac{1}{2}\xi\Lambda_{1J} + \tfrac{1}{2}\eta\Lambda_{2J} + \xi\eta\Gamma_J) \\ &(\tfrac{1}{2}\Lambda_{1M} + \eta\Gamma_M)(\tfrac{1}{2}\Lambda_{2N} + \eta\Gamma_N)\,d\xi\,d\eta \end{aligned} \tag{7.9}$$

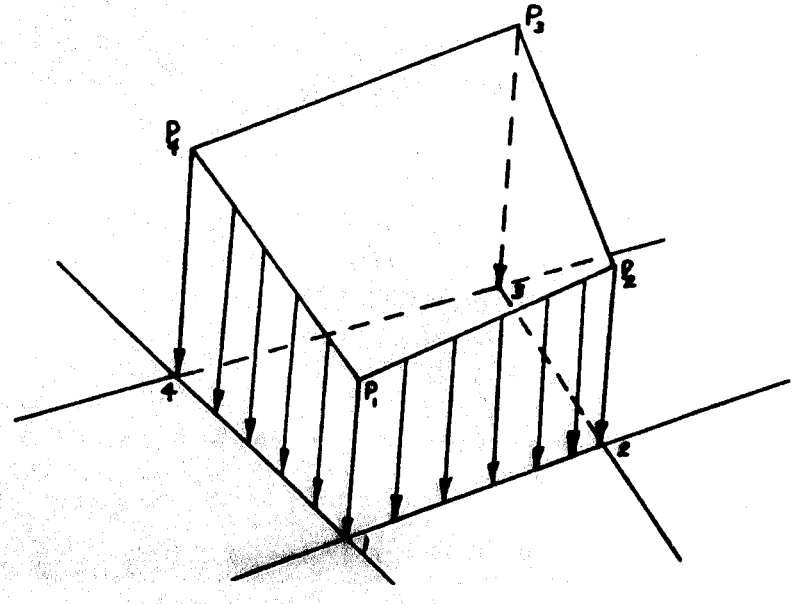


Figure 7.1. Definition of a Pressure Boundary Condition Along an Element Face

Integrating yields

$$f_{iI} = -\tfrac{1}{64}\, p_J\, e_{ijk}\, x_{jM}\, x_{kN}[(\Sigma_I\Sigma_J + \tfrac{1}{3}\Lambda_{1I}\Lambda_{1J} + \tfrac{1}{3}\Lambda_{2I}\Lambda_{2J} + \tfrac{1}{9}\Gamma_I\Gamma_J)\Lambda_{1M}\,\Lambda_{2N} + (\tfrac{1}{3}\Sigma_I\Lambda_{1J} + \tfrac{1}{3}\Lambda_{1I}\Gamma_J + \tfrac{1}{9}\Gamma_I\Lambda_{2J} + \tfrac{1}{9}\Lambda_{2I}\Gamma_J)\Lambda_{1M}\,\Gamma_N + (\tfrac{1}{3}\Sigma_I\Lambda_{2J} + \tfrac{1}{3}\Lambda_{2I}\Gamma_J + \tfrac{1}{9}\Gamma_I\Lambda_{1J} + \tfrac{1}{9}\Lambda_{1I}\Gamma_J)\Gamma_M\Lambda_{2N}] \quad (7.10)$$

The above expression may be evaluated to yield the following formula for calculating the nodal forces:

$$\begin{bmatrix} F_{i1} \\ F_{i2} \\ F_{i3} \\ F_{i4} \end{bmatrix} = \tfrac{1}{72}[(x_{j3}-x_{j1})(x_{k4}-x_{k2}) + (x_{j2}-x_{j4})(x_{k3}-x_{k1})] \begin{bmatrix} 2(2p_1+p_2+p_4)+p_3 \\ 2(2p_2+p_3+p_1)+p_4 \\ 2(2p_3+p_4+p_2)+p_1 \\ 2(2p_4+p_1+p_3)+p_2 \end{bmatrix}$$

$$+ \tfrac{1}{72}[(x_{j2}-x_{j1})(x_{k3}-x_{k4}) + (x_{j3}-x_{j4})(x_{k1}-x_{k2})] \begin{bmatrix} -(2p_1+p_4) \\ (2p_2+p_3) \\ (2p_3+p_2) \\ -(2p_4+p_1) \end{bmatrix}$$

$$+ \tfrac{1}{72}[(x_{j1}-x_{j4})(x_{k3}-x_{k2}) + (x_{j3}-x_{j2})(x_{k4}-x_{k1})] \begin{bmatrix} -(2p_1+p_2) \\ -(2p_2+p_1) \\ (2p_3+p_4) \\ (2p_4+p_3) \end{bmatrix} \quad (7.11)$$

where the indices $\{i,j,k\}$ form a negative permutation. Note that a positive pressure gives forces directed inward.

The nodal values for the pressure are calculated using the user-supplied scale factor and time history function. The values are recalculated every CG iteration.

7.2.2 Nodal Forces

Nodal point external forces are simply applied by calculating the magnitude of the force determined by the user-supplied scale factor and a time history function. The time history function is evaluated at the beginning of the time step.

7.2.3 Gravity or Body Forces

Gravity or body forces are computed with the evaluation of the second integral in Equation 3.6. This is done as follows:

$$\int_{V_e} \rho b_i \delta\dot{u}_i\, dV = \rho b_{iI} V_e \delta_{IJ} \delta\dot{u}_{iJ} \quad . \quad (7.12)$$

where V_e is the element volume and δ_{IJ} is the Kroneker delta.

Body forces are input by the use of a specified time history function for each component. For example, a gravity load or body force in the z-direction would only use the Z GRAVITY FUNCTION option. General body forces that vary with position and time may be input from a file using the DISTRIBUTED LOADS option.

Blank Page

8. Verification and Sample Problems

Sample problems are included for code verification and to acquaint the user with the use of the JAC3D program. The first section gives problems intended to verify the general coding of JAC3D. The second section presents verification studies specific to the isothermal elastic/plastic material model.

8.1 Elastic Problems for Verification of Continuum and Truss Elements

Several elastic verification problems are presented. These include the large-displacement analysis of a thin beam with and without truss elements, the rotation of a unit cube, and the pressurization of the internal surface of a sphere.

8.1.1 Cantilever Beam

The large deformation of an elastic cantilever beam is included for comparison with the analytical solution as formulated by Holden [32]. The beam problem is challenging for the CG method, which has a great deal of difficulty finding a solution. The beam has a length-to-thickness ratio of 30 and, to simulate plane stress conditions, Poisson's ratio is set equal to zero. Gravity and normal pressure loading conditions are presented.

First, the beam is loaded with gravity, which keeps the direction of load constant throughout the analysis. Following the notation and development of Holden, the equation for the slope of the beam is

$$\frac{d^2\theta}{d\bar{s}^2} = -k\bar{s}\cos\theta\,, \tag{8.1}$$

where θ is the angle between the beam neutral axis and the x-axis, $\bar{s} = s/L$ is the normalized arc length along the beam neutral axis, $k = wL^3/EI$ is a nondimensional loading parameter, L is the length of the beam, w is the loading intensity (load per unit length), E is Young's modulus, and I is the beam's moment of inertia. This equation describes the finite deflection of uniform beams using the Euler-Bernoulli law of bending subject to vertical (gravity) loading. Boundary conditions for a cantilever beam are

$$\frac{d\theta}{d\bar{s}} = 0 \quad \text{at} \quad \bar{s} = 0 \text{ (free end)} \tag{8.2}$$

$$\theta = 0 \quad \text{at} \quad \bar{s} = 1 \text{ (fixed end)} \tag{8.3}$$

The normalized horizontal and vertical deflections of the free end of the beam are then

given by

$$h/L = \int_0^1 \cos\theta \, d\bar{s} \tag{8.4}$$

and

$$\delta/L = \int_0^1 \sin\theta \, d\bar{s} \,, \tag{8.5}$$

respectively. Equation 8.1 was solved using a Runge-Kutta procedure, the integrations for deflections were computed using adaptive quadrature, and the results checked by comparison to Holden's published solution.

The finite element model, shown in Figure 8.1, has thirty elements along its length, four through the bending direction, and one transverse to the bending. The nonlinear response was calculated for the gravity-loaded case and is compared to the beam-theory solution in Figure 8.2. The comparison for this case is very good, although because of numerical difficulty the JAC3D solution terminated at $k = 17$. In Figure 8.1 the deformed shape of the mesh for loads corresponding to $k = 6.5$ and $k = 17$ is shown in comparison to the undeformed mesh.

Convergence of the solution for the beam problem is very slow. First, the spread of eigenvalues in the problem is large, and any indirect interative solution technique with a diagional conditioning matix will have difficulty. If the problem is ill-conditioned in the linear approximation, as in this case, then adding the nonlinearity of large deformation (i.e., taking into account the rotation of the beam) compounds the difficulty. To lower the impact of the nonlinearities, the calculation is begun by first solving the initial load step assuming linear geometry. Then, using the linear results as a starting vector, the geometric-nonlinear effects are included and the first step is recalculated (TRIAL LINEAR). The starting displacement increment for each of the following time steps is taken to be the incremental displacement calculated in the previous step (TRIAL VELOCITY FUNCTION). For the gravity-loaded case, the load steps took an average of 580 nonlinear interations each. Load increments of $k = 0.0648$ were used, with a convergence tolerance of 0.03 on the residual force norm of Equation 3.16. To get the correct bending response, the default value of the HOURGLASS PARAMETER must be used.

The problem was also analyzed with pressure applied along the top of the beam, so that the load remained normal to the surface throughout the deformation. The beam-theory equation for this case is

$$\frac{d^2\theta}{d\bar{s}^2} = -k\bar{s} \,, \tag{8.6}$$

with the same boundary conditions as before. For large load magnitudes, this configuration causes more severe bending of the beam. The input file for the pressure-loaded case is shown in Figure 8.3. Once again, the analytic solution is compared to the JAC3D calculation in Figure 8.4, and the deformed shape of the beam at several load levels is shown in Figure 8.5. In this case, the finite element model is somewhat stiffer than Euler-Bernoulli beam theory predicts, particularly at the higher loads. This is probably

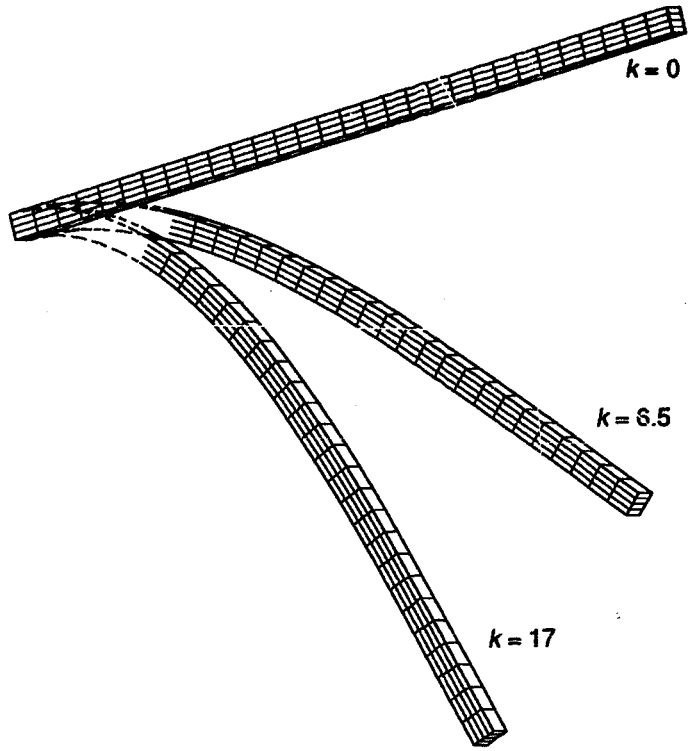


Figure 8.1. Finite Element Model for Beam Problem.

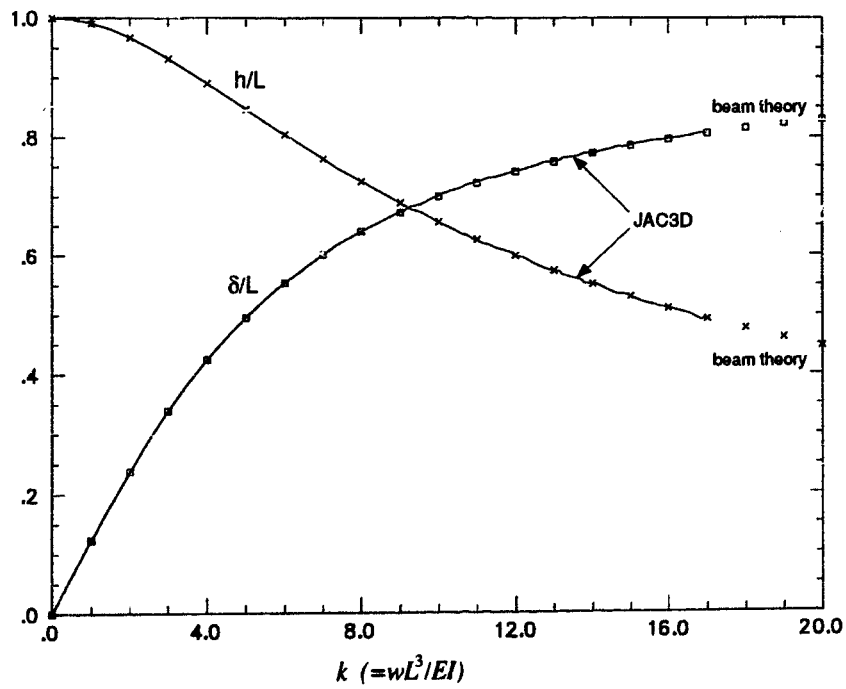


Figure 8.2. Comparison of Displacements for a Beam with Gravity Loading.

```
TITLE
  ELASTIC BEAM WITH PRESSURE LOADING
TRIAL LINEAR
MAXIMUM ITERATIONS, 3000
ITERATION PRINT,      10
RESIDUAL TOLERANCE, 0.03
MAXIMUM TOLERANCE,  0.06
CGRESET LIMITS, 790, 80
TRIAL VELOCITY FUNCTION, 2
SOLUTION FUNCTION,       3
FUNCTION 1 $ PRESSURE LOADING
  0.0, 0.0
  2.0, 2.0
END
FUNCTION 2 $ TRIAL VELOCITY
  0.0, 1.0
  2.0, 1.0
END
FUNCTION 3 $ SOLUTION
  0.0,  310
  1.55
END
PRESSURE, 5, 1, 400.
X DISPLACEMENT, 1
Y DISPLACEMENT, 1
Z DISPLACEMENT, 1
Z DISPLACEMENT, 3
Z DISPLACEMENT, 4
MATERIAL, 1
   ISOTHERMAL ELASTIC PLASTIC
   YOUNGS MODULUS,1E+7
END
EXIT
```

Figure 8.3. Input for the Pressure-Loaded Beam Problem.

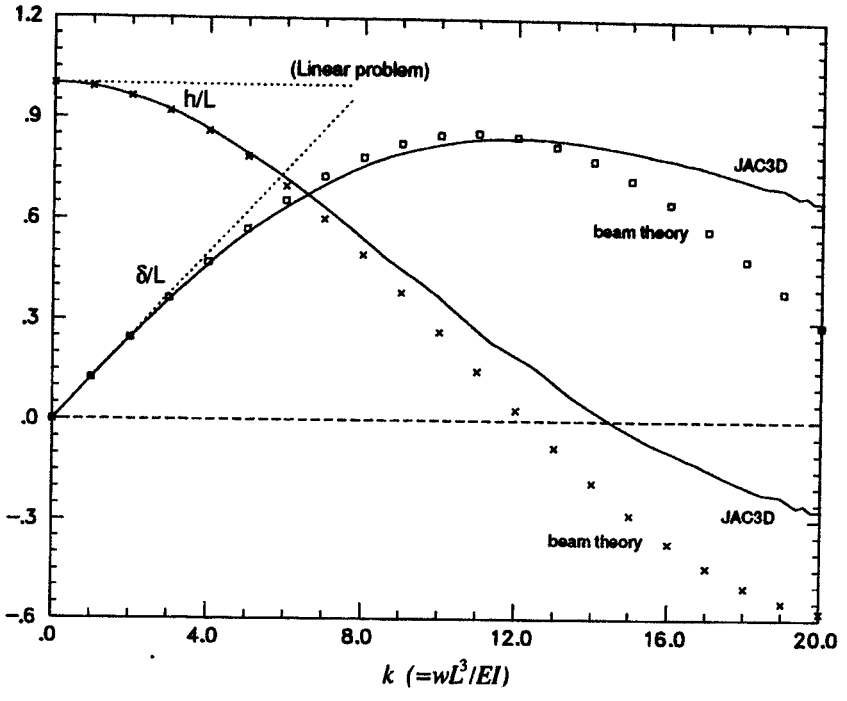


Figure 8.4. End Displacement for a Beam with Uniform Pressure Loading.

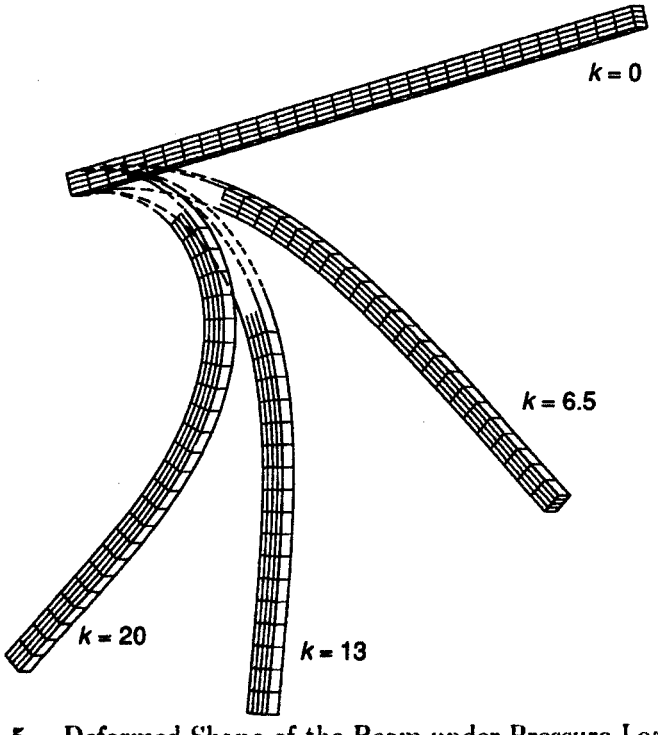


Figure 8.5. Deformed Shape of the Beam under Pressure Loading.

due to the fact that when the beam starts bending back on itself, the radius of curvature is no longer large compared to the thickness of the beam. JAC3D had a somewhat easier time with this load configuration: it ran all the way to completion at $k = 20$ and took an average of 365 nonlinear iterations per load step. As a point of reference, the JAC3D solution with linear geometry assumptions (LINEAR PROBLEM) is also plotted in Figure 8.4. This solution agrees with the linear beam theory prediction of $\delta/L = k/8$.

Truss elements were added along the top and bottom row of elements to increase the bending stiffness by a factor of two. The addition of truss elements produced the same results as stiffening the material by a factor of two, thus verifying the coding for the truss elements. To further verify the coding of the truss and continuum elements, the beam was rotated to an arbitrary position in 3-space and the pressure-loading calculation was repeated. The results were insensitive to the change in orientation.

8.1.2 Unit Cube

A unit cube as shown in Figure 8.6 is first loaded with uniaxial pressure and then rotated about one corner. The input to the unit cube problem is shown in Figure 8.7. The INITIAL EQUILIBRIUM capability is exercised to obtain the uniaxial stress state before rotation begins. CGRESET LIMITS are chosen so as to help the single-element problem converge on initial loading. A rotation of 90° is then accomplished in 10 steps. The stress

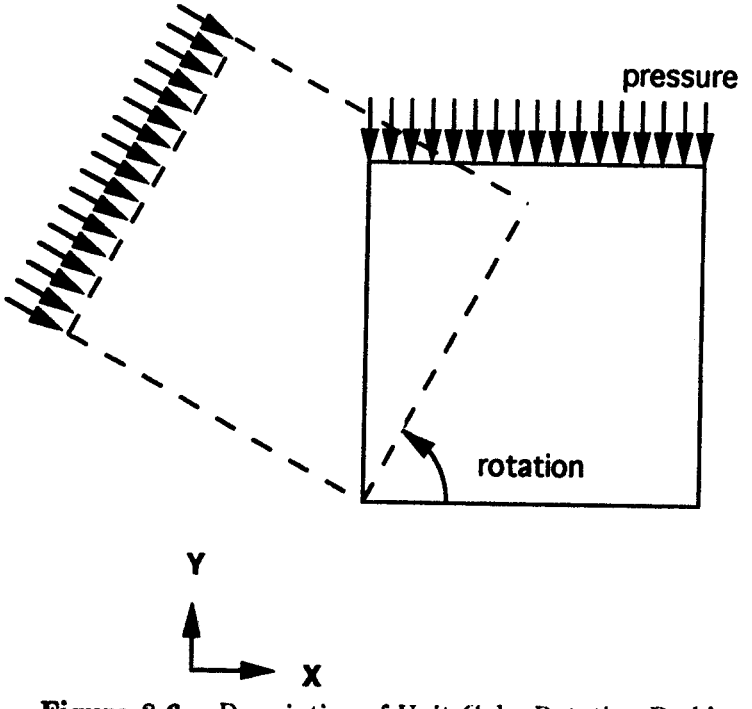


Figure 8.6. Description of Unit Cube Rotation Problem.

```
TITLE
 ONE ELEMENT ROTATION TEST
HOURGLASS PARAMETER 2
INITIAL EQUILIBRIUM
CGRESET LIMITS 0, 12, 10
SOLUTION FUNCTION 1
OUTPUT FUNCTION 1
FUNCTION 1
 0    10
 4.0
END
RESIDUAL TOLERANCE = .00001
MAXIMUM ITERATIONS = 100
ITERATION PRINT = 1
MATERIAL,1,1
YOUNGS MODULUS 1.0E6
END
PRESSURE 5,4,1
X DISPLACEMENT 4 ,2 ,1.
Y DISPLACEMENT 4 ,3 ,1.
X DISPLACEMENT 3
Y DISPLACEMENT 3
Z DISPLACEMENT 5
Z DISPLACEMENT 6
FUNCTION,4
 0 10000
 4 10000
END
FUNCTION,2
 0.0000000, 0.0000000
 0.4000000, 0.0123117
 0.8000000, 0.0489436
 1.2000000, 0.1089938
 1.6000000, 0.1909835
 2.0000000, 0.2928941
 2.4000000, 0.4122159
 2.8000000, 0.5460110
 3.6000000, 0.8435554
 4.0000000, 0.9999833
END
FUNCTION,3
 0.0000000, 0.0000000
 0.4000000, 0.1564346
 0.8000000, 0.3090174
 1.2000000, 0.4539911
 1.6000000, 0.5877860
 2.0000000, 0.7071076
 2.4000000, 0.8090178
 2.8000000, 0.8910072
 3.2000000, 0.9510553
 3.6000000, 0.9876868
 4.0000000, 1.0000000
END
EXIT
```

Figure 8.7. Input for Unit Cube Rotation Problem.

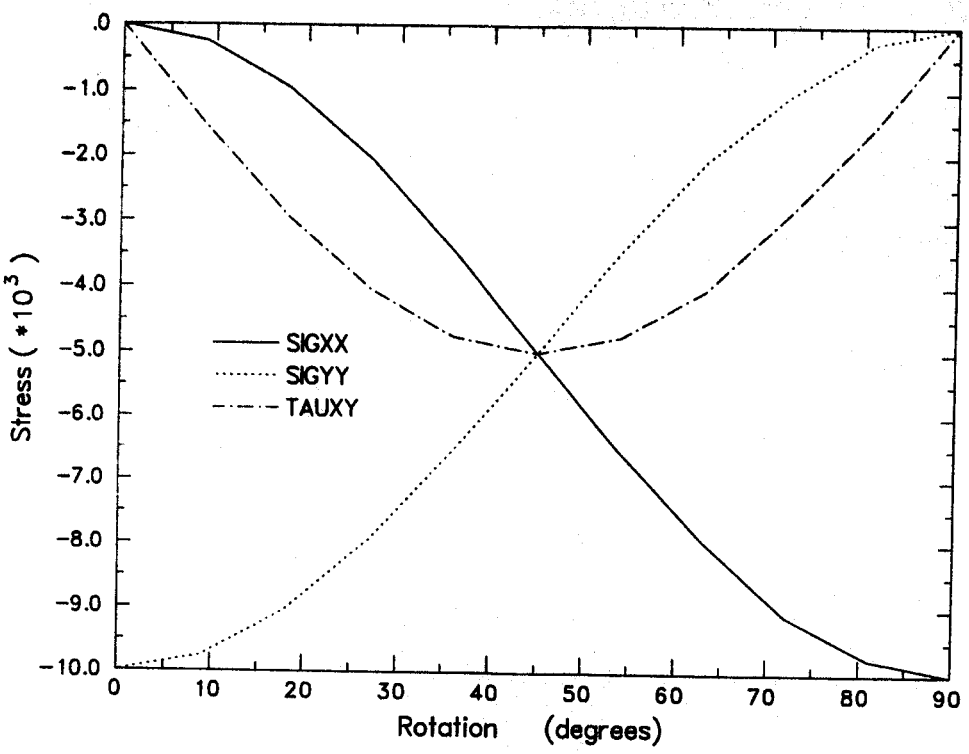


Figure 8.8. Stress Results for Unit Cube Rotation Problem.

state as a function of rotation is shown in Figure 8.8. The rotation algorithm is very accurate. In fact, for this problem it is essentially exact.

8.1.3 Internally-Pressurized Sphere

A spherical geometry was chosen to demonstrate the performance of the nonlinear CG method and to verify the coding of JAC3D. With a spherical geometry, relatively simple loading conditions can be used to exercise a significant portion of the code. For example, all the components of displacement, stress, strain, and material state variables are used. The sphere is layered so that multiple material models can be utilized. Interface conditions are applied to verify the contact surface algorithm for fixed interfaces. Loading and response is one-dimensional (radial), so that a small sector of the sphere may be analyzed to obtain a high-resolution solution. The model can then exercise skew-displacement boundary conditions and the sliding interface algorithm.

An outside-to-inside radius ratio of two was chosen for the analyses. Two different finite element meshes were used, which will be referred to as the "octant" and "sector" models, respectively. The models, which both use 40 elements though the thickness, are shown in Figures 8.9 and 8.10. The octant model has four material layers joined by fixed interfaces, and symmetry boundary conditions are applied on the three orthogonal coordinate planes. The mesh has 1,840 elements and 2,596 nodes. The sector model

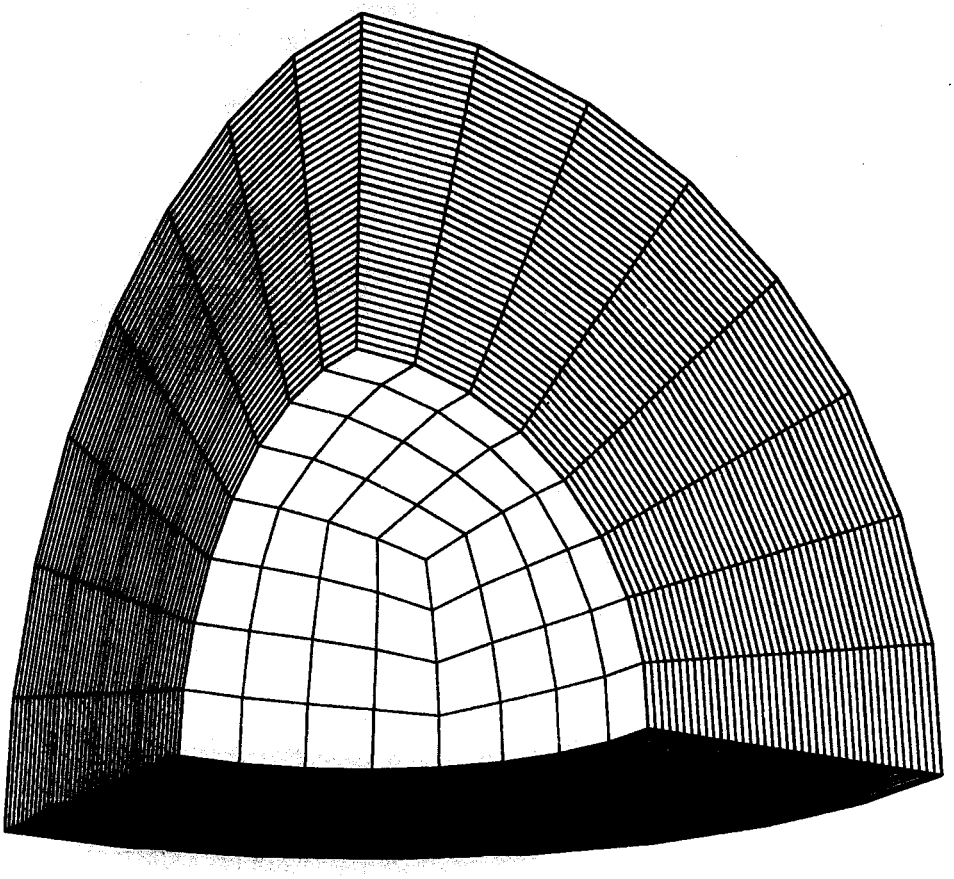


Figure 8.9. Octant Finite Element Model for Sphere Problem.

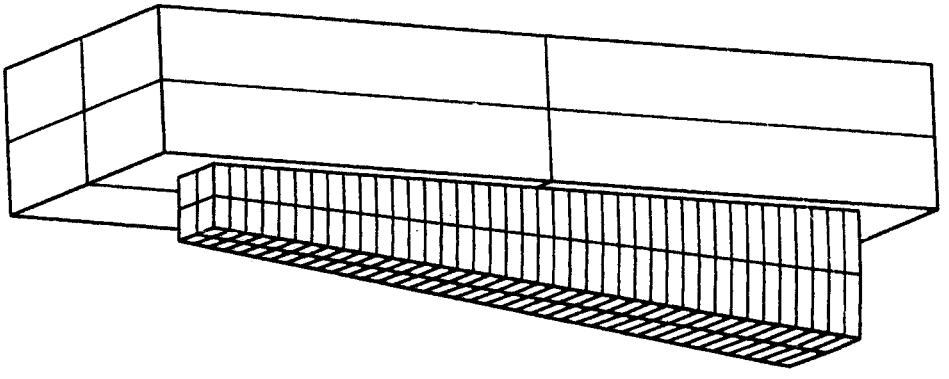


Figure 8.10. Sector Finite Element Model for Sphere Problem.

covers a 5° sector of the sphere with a single material block. The first two sides are aligned with coordinate planes and have displacement constrained in the y- and z-directions, respectively. A third side has normal displacement constrained by use of the **XYPLANE DISPLACEMENT** option. The fourth side is constrained to slide along a rigid surface placed above the sector. The sector mesh has 168 elements and 396 nodes.

There is a simple analytic solution to this problem. The elastic solution given by Mendelson [33, Equation 8.3.11] is

$$u = \left[(1-2\mu)r + \frac{(1+\mu)b^3}{2r^2}\right] \frac{p}{E(b^3/a^3 - 1)}, \tag{8.7}$$

where E and μ are the elastic constants, r is the radial position, a and b are inside and outside radii, respectively, and p is the applied internal pressure. To obtain a geometrically-nonlinear solution, this equation was applied iteratively, with the radial positions updated each iteration to reflect the displacements calculated in the previous iteration.

The radial displacement calculated by JAC3D is compared to the analytic solution in Figure 8.11. Using a Young's modulus of 1.0×10^7 and a Poisson's ratio of 0.3 with an applied pressure of 10,000 resulted in an internal deformation of 0.040034 using the sector model, which compares favorably to the analytical solution of 0.040049. JAC3D required

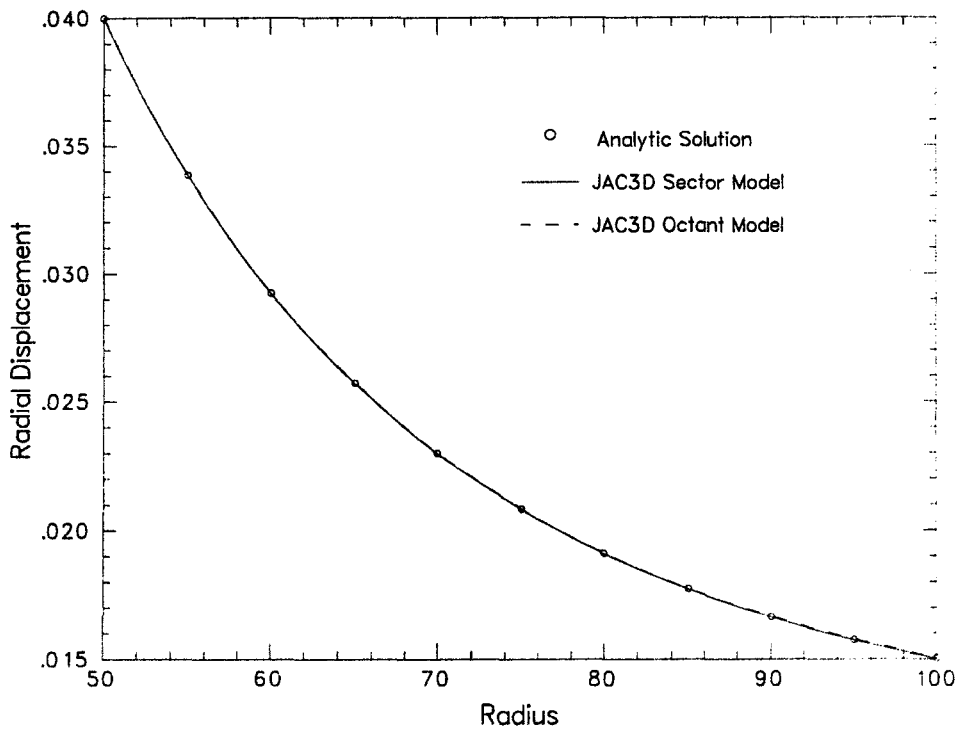


Figure 8.11. Radial Displacement for the Pressurized Linear Elastic Sphere Problem.

119 iterations (17 CPU seconds) for the sector model to reach a convergence tolerance of 0.001. The input commands for this problem are shown in Figure 8.12. Using the octant model JAC3D required 305 iterations (26 CPU seconds), and calculated a final internal radial displacement of 0.040088. This solution is plotted with a dashed line in Figure 8.11, and is barely distinguishable from that of the sector model.

```
TITLE
 ELASTIC SPHERE WITH PRESSURE LOADING
ITERATION PRINT = 10
MAXIMUM ITERATIONS = 600
RESIDUAL TOLERANCE = .001
SOLUTION FUNCTION = 1
OUTPUT FUNCTION = 1
FUNCTION = 1  $ SOLUTION
   0.0,1
   1.0
END
FUNCTION = 2  $ PRESSURE LOADING
   0.0,0.0
   1.0,10000.0
END
PRESSURE = 1,2,1.0
X DISPLACEMENT = 1       $
Y DISPLACEMENT = 1       $  Rigid surface for sliding interface
Z DISPLACEMENT = 1       $
Y DISPLACEMENT = 2
Z DISPLACEMENT = 3
XYPLANE DISPLACEMENT = 4, 95.0
SLIDING SURFACE = 11,10, 0.0, 1.E-4
MATERIAL 1
ISOTHERMAL ELASTIC PLASTIC
   YOUNGS MODULUS = 1.0E7
   POISSONS RATIO = 0.3
END
EXIT
```

Figure 8.12. Input for Sector Model, Pressurized Sphere Problem.

8.2 Isothermal Elastic/Plastic Verification Problems

8.2.1 Rotating Ring

The rotating ring problem, shown in Figure 8.13, is included to verify the large-rotation capability for elastic/plastic material reponse. The ring is given a slow in-plane rotation and internally pressurized beyond the yield stress of the material. It is positioned in the xy-plane with its center at the origin. The initial inside and outside radii are 0.95 and 1.05, respectively. The out-of-plane thickness is 0.1, and plane strain conditions are imposed by restricting displacements to the xy-plane. The ring material has a Young's modulus of 10.3×10^6, a Poisson's ratio of $\frac{1}{3}$, a yield stress of 4.15×10^4, a hardening modulus of 5.17×10^5, and a hardening parameter β of 1.0. During the pressure loading sequence the ring was rotated by applying opposing horizontal displacements to a pair of nodal points on the top and bottom of the ring, respectively, and opposing vertical displacements to nodes on the left and right sides, respectively. Figure 8.13 shows the ring before and after loading, with Element 1 shaded in both cases for reference. The input for the ring problem is shown in Figure 8.14.

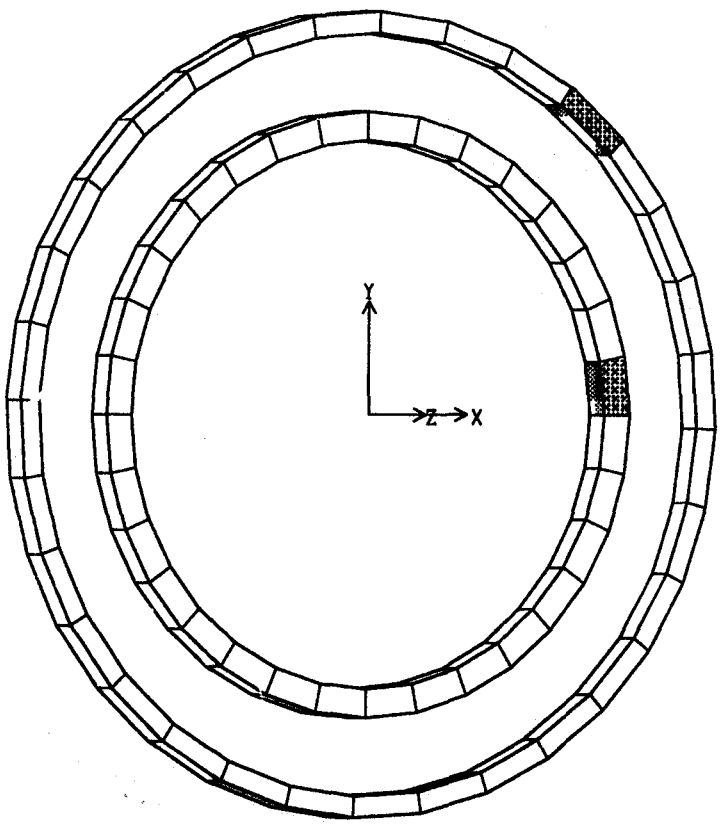


Figure 8.13. Geometry for the Rotating Ring Problem.

```
TITLE
  EXPANDING AND ROTATING RING
MAXIMUM ITERATIONS 1000
ITERATION PRINT 10
GENESIS OUTPUT
MATERIAL 1, 1
  YOUNGS MODULUS      10.3+6
  POISSONS RATIO       0.333333333
  YIELD STRESS         4.15+4
  HARDENING MODULUS    5.17+5
  HARDENING PARAMETER 1.0
END
SOLUTION FUNCTION 5
OUTPUT FUNCTION 4
TRIAL VELOCITY FUNCTION 3
FUNCTION 1        $ LOADING
  0.0          0.0
  1.0       4000.0
  2.0       5000.0
  2.5       6000.0
 10.0      10000.0
 20.0      15000.0
END
FUNCTION 2        $ DISPLACEMENT ROTATION FUNCTION
  0.0          0.0
 20.0          1.055
END
FUNCTION 3        $ TRIAL DISPLACEMENT
  0.0          0.0
  4.0          0.0
  4.25         1.0
 20.0          1.0
END
FUNCTION 4        $ OUTPUT
  0.0          9.0
 18.0
END
FUNCTION 5        $ SOLUTION FUNCTION
  0.0          1.0
  1.0          1.0
  2.0          8.0
  2.5         30.0
 10.0         40.0
 18.0
END
X DISPLACEMENT 4 2 1
Y DISPLACEMENT 3 2 -1
X DISPLACEMENT 2 2 -1
Y DISPLACEMENT 1 2 1
Z DISPLACMENT 5
Z DISPLACMENT 6
PRESSURE 10 1 1.0
RESIDUAL TOLERANCE 0.005
EXIT
```

Figure 8.14. Input for the Rotating Ring Problem.

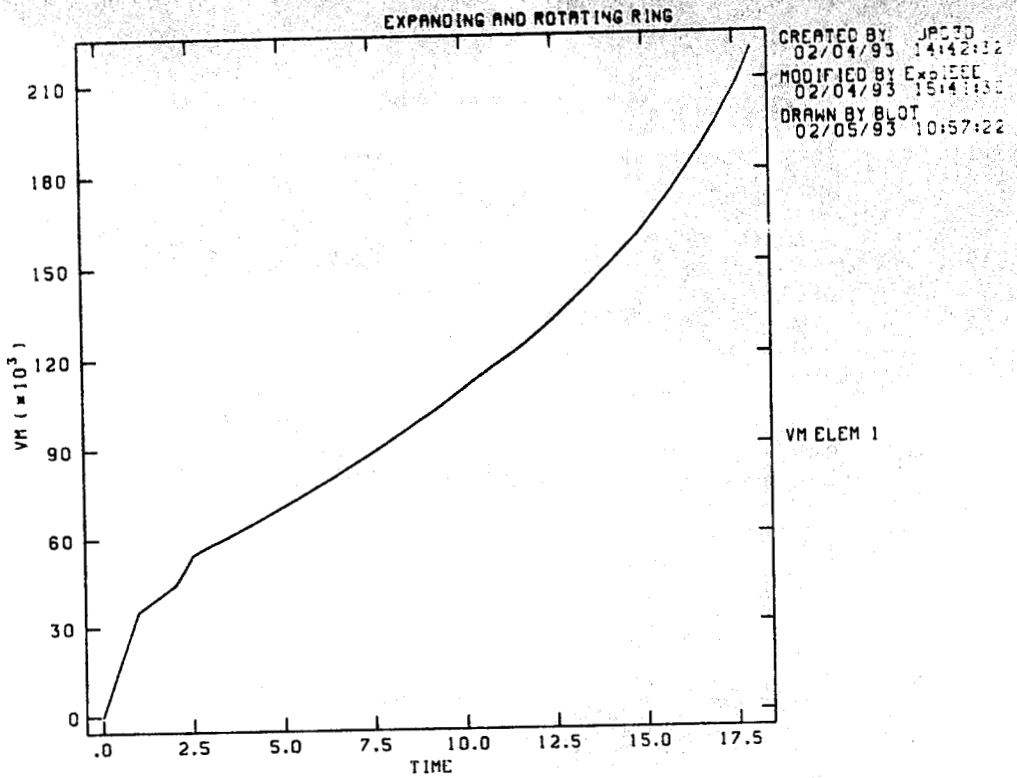


Figure 8.15. Effective Stress Response of the Rotating Ring.

The effective stress as a function of time is depicted in Figure 8.15. Note that the abrupt changes in slope of the curve reflect the change in the applied pressure rate. The total rotation is approximately 45° and all the elements exhibit identical response to within ±0.2 percent. The effective stress calculated is the same as it would have been had the ring not been rotated.

8.2.2 Hollow Sphere

Elastic/plastic analyses of the hollow sphere discussed in Section 8.1.3 were performed to verify the isothermal elastic/plastic material model. A yield stress of $\sigma_y = 10,000$ was used for all the analyses.

The first problem considers the material to be perfectly plastic. When enough pressure is applied, yielding begins on the inside surface. As the pressure is increased further, the plastic zone expands radially until the entire sphere has yielded. The radius c of the elastic/plastic interface is related to the applied pressure p by the following equation given by Mendelson [33, Equation 8.3.5]:

$$P = 2\ln\rho_c + \frac{2}{3}\left(1 - \frac{1}{\beta_c^3}\right) \tag{8.8}$$

Here the dimensionless variables are $P = p/\sigma_y$, $\rho_c = c/a$, and $\beta_c = b/c$. Once the

interface radius is computed, the dimensionless effective stress $S \equiv |\sigma_\theta - \sigma_r|/\sigma_y$ in the outer elastic region reduces to simply c^3/r^3. By definition the dimensionless effective stress is unity in the plastic region.

For the finite element solution to this problem, internal pressure was applied first to the octant model and the material was considered to be perfectly plastic. Sufficient pressure was applied on the first load step to initiate yielding on the inside surface (letting $c = a$ in Equation 8.8). The pressure was then increased in 20 equal load steps to the point that the entire sphere should have yielded ($c = b$). The effective stress along a radial line from an element on the inside surface to an element on the outside surface for load steps 1, 6, 11, 16, and 21 are shown in Figure 8.16, where it is compared to the analytic solution. A convergence tolerance of 0.001 was specified in JAC3D. Note that for a perfectly plastic material, there is no equilibrium solution when internal pressure causes the entire sphere to yield. (The balloon pops!) Thus, on the last load step JAC3D did not converge to the specified equilibrium tolerance, although the stresses resulting from the non-equilibrium final state indicate that the sphere had indeed completely yielded. The first twenty load steps required a total of 4,118 iterations for the octant model.

The analysis was repeated using the sector model, shown with x's in Figure 8.16. This idealization produced slightly higher effective stresses, particularly at the later load

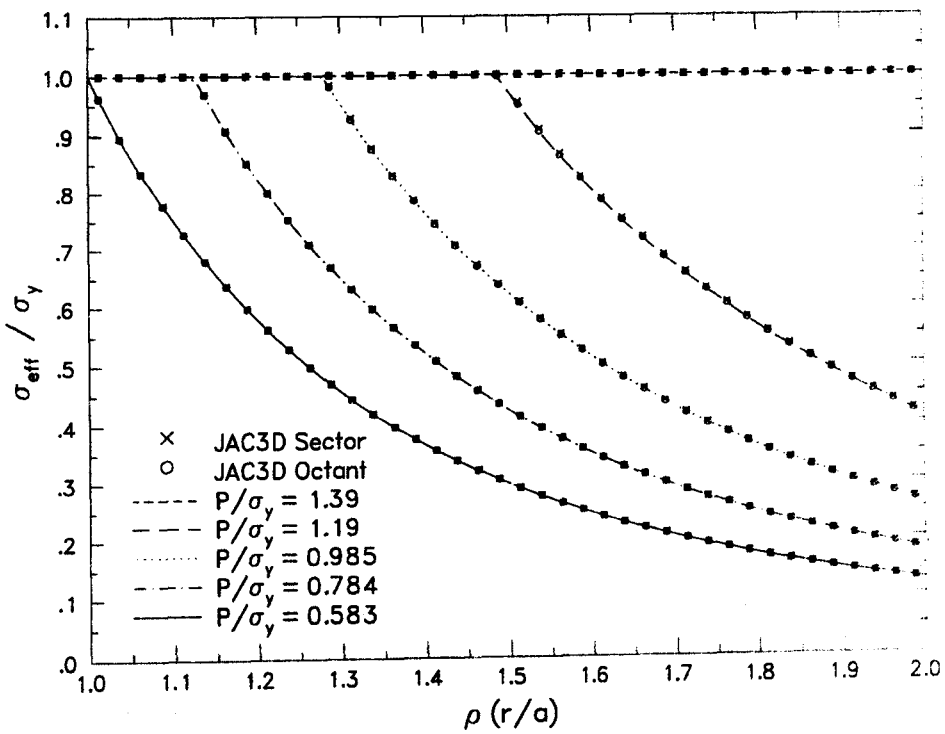


Figure 8.16. Effective Stress Distribution for the Pressurized Elastic/Perfectly-Plastic Sphere Problem.

steps. It took 2,444 iterations to solve the first twenty load steps using the sector model. Both JAC3D analyses predicted higher stresses than the analytic solution, more noticeable at the higher pressures. This is probably due to the fact that the analytic solution assumes linear geometry, whereas the JAC3D analyses were geometrically nonlinear.

The next problem includes linear strain hardening in the material plasticity. The radius of the elastic/plastic interface for this case is computed from [33, Equation 8.6.20]

$$P = \frac{\frac{4}{3}m(1-\mu)(1-1/\beta^3)\rho_c^3 + 2(1-m)\ln\rho_c + \frac{2}{3}(1-m)(1-1/\beta_c^3)}{1-m+2m(1-\mu)} \tag{8.9}$$

where m is the ratio of the hardening modulus to Young's modulus, μ is Poisson's ratio, and $\beta = b/a$. Note that for $m = 0$, this reduces to the perfectly-plastic case. The effective stress in the elastic region is the same as before, whereas in the plastic region [33, Equation 8.6.22][1]

$$S = \frac{1-m+2m(1-\mu)c^3/r^3}{1-m+2m(1-\mu)} . \tag{8.10}$$

This problem was analyzed using the octant model with a strain-hardening modulus of 1.0×10^6 ($m = 0.1$). Because of the strain-hardening, a higher final pressure was required to yield the entire sphere. Moreover, because the material continues to harden with increasing strain, the last load step was stable and converged easily. The results are shown in Figure 8.17. For the later load steps the finite element solution predicts slightly higher stresses than the analytic solution. Once again this is likely due to the geometrical nonlinearity accounted for in the JAC3D solution. For comparison, the first twenty load steps in this problem required a total of 3,914 iterations.

To verify the use of thermal loads in the JAC3D program, the sphere was analyzed using a sequence of steady-state thermal loads obtained by raising the temperature of the inside surface while the outside surface temperature was held at zero. The analytic solution for this problem is taken from Section 5.4 of Reference [34], with notation changed slightly for compatibility with the foregoing discussion. Denoting the inside surface temperature by T_0 and the coefficient of thermal expansion by α, a dimensionless thermal load is defined as

$$\tau \equiv \frac{E\alpha T_0}{\sigma_y(1-\mu)} \tag{8.11}$$

Yielding begins at the inner surface due to hoop compression and expands outward. The radius c of the initial elastic/plastic interface is found by solving

$$\tau = 2\left(\frac{c}{a} - \frac{c}{b}\right)\left[\frac{1-c^3/b^3+\ln(c^3/a^3)}{(2+c/b)(1-c/b)^2}\right] \tag{8.12}$$

Letting $\alpha = 0.00001$ and using the previous elastic/perfectly-plastic material properties and geometry, initial yield at $c = a$ occurs at $T_0 = 98.0°$. Since there is no strain

[1]The expression for S_ϕ in the referenced equation is missing the term $+\frac{1-m}{1-m+2m(1-\mu)}$.

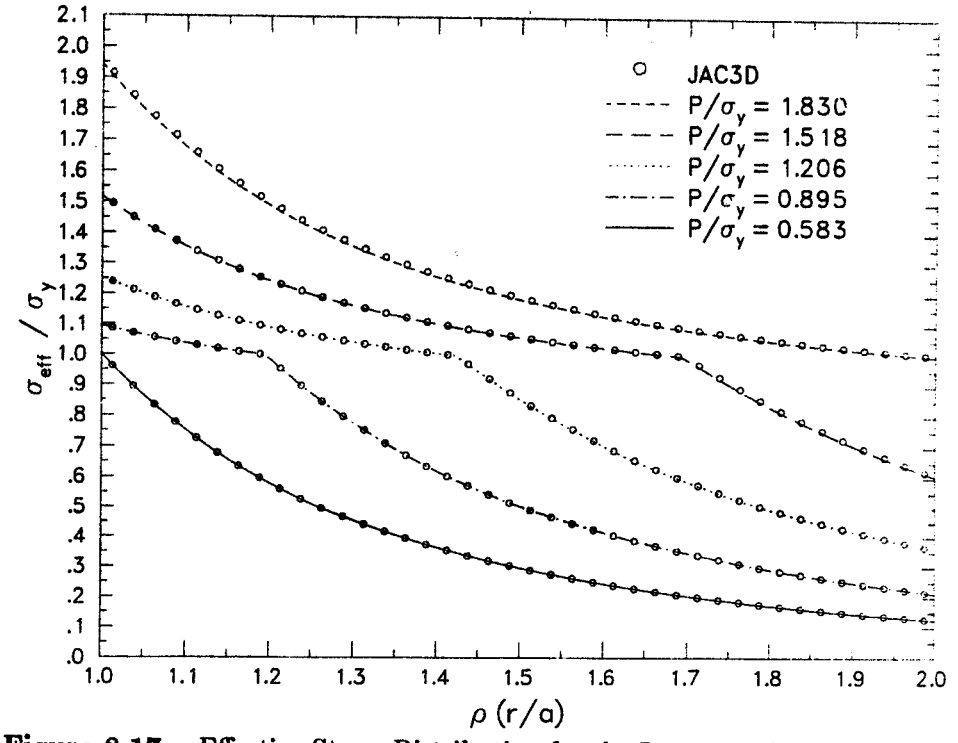


Figure 8.17. Effective Stress Distribution for the Pressurized Elastic/Plastic Strain-Hardening Sphere Problem.

hardening, the dimensionless effective stress S in the plastic region remains at unity. In the elastic region, S is given by

$$S = \left| B\frac{3b^3}{2r^3} - \tau\frac{ab}{2r(b-a)} \right| \tag{8.13}$$

where

$$B = 2\frac{c^3}{b^3}\left[\frac{1 - c/b + \ln(c/a)}{(2 + c/b)(1 - c/b)^2}\right] \tag{8.14}$$

When the plastic boundary has advanced to a radius c_1, a second plastic zone is initiated at the outer surface due to hoop tension. The radius c_1 is found by solving

$$\ln\frac{c_1}{a} = \frac{2b}{3c_1}\left(1 - \frac{c_1}{b}\right)^2 \tag{8.15}$$

For the present problem, c_1/a has a value of 1.197 and occurs at an inner surface temperature (from Equation 8.12) of 264.8°. Above this temperature, the second plastic zone spreads inward from the outer surface while the inner plastic zone continues to spread outward. When there are two separate plastic zones, Equation 8.12 no longer holds.

Instead we must simultaneously solve for both elastic/plastic boundaries, using

$$\tau = 2\left(\frac{d}{a} - \frac{d}{b}\right)\left(1 + \frac{c^2/d^2}{1 - c/d}\right) \tag{8.16}$$

$$\ln\left(\frac{cd}{ab}\right) = \frac{2c}{3d}\left(\frac{d}{c} - 1\right)^2 \tag{8.17}$$

where d is the radius to the boundary of the outer plastic region. The stresses in the elastic region are obtained from Equation 8.13 with

$$B = 2\frac{c^3}{b^3}\left[\frac{d/c}{3(1 - c/d)}\right] \tag{8.18}$$

For the JAC3D calculation, the initial load step corresponded to $T_0 = 98°$. Solutions were then calculated using the octant model for 20 equal steps in which T_0 was progressively increased to 600°. The resulting effective stress vs. radial position at load steps 1, 6, 11, 16, and 21 is shown in Figure 8.18, compared to the analytic solutions. As with the previous two solutions, the agreement is very satisfactory. To reach a convergence tolerance of 0.001 required 2,737 total iterations.

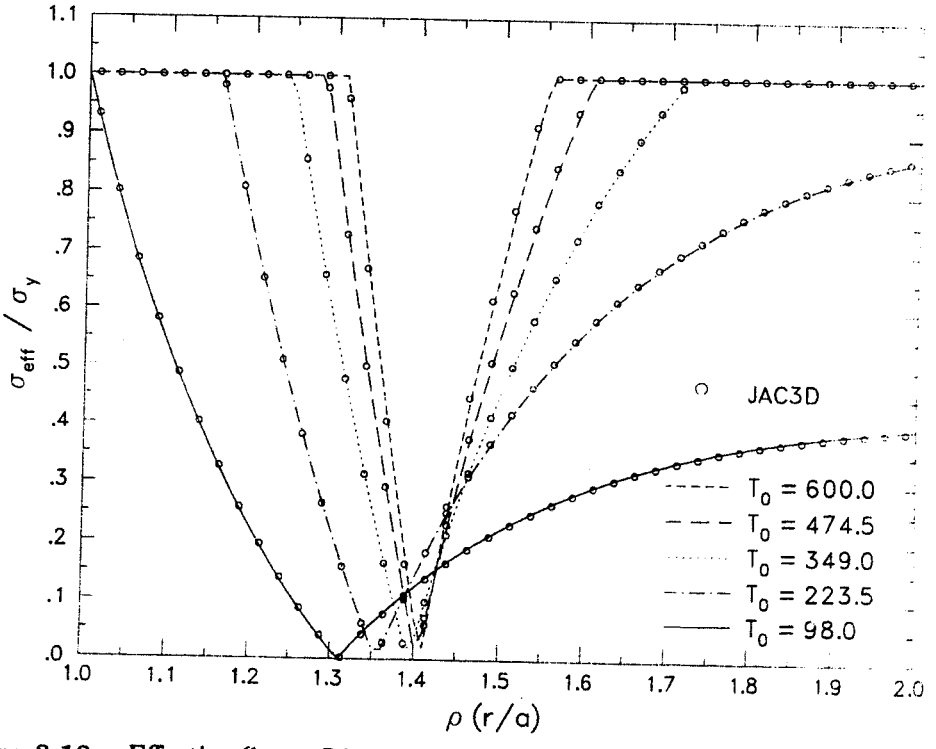


Figure 8.18. Effective Stress Distribution for the Thermally-Loaded Elastic/Perfectly-Plastic Sphere Problem.

References

[1] J. H. Biffle, "JAC—A Two-Dimensional Finite Element Computer Program for the Nonlinear Response of Solids with the Conjugate Gradient Method," SAND81-0998, Sandia National Laboratories, Albuquerque, NM, 1981. (HQS.880517.2257)

[2] S. W. Key, Z. E. Beisinger, and R. D. Krieg, "HONDO II—A Finite Element Computer Program for the Large Deformation Dynamics Response of Axisymmetric Solids," SAND78-0422, Sandia National Laboratories, Albuquerque, NM, October 1978. (NNA.920512.0110)

[3] J. H. Biffle and M. H. Gubbels, "WULFF—A Set of Computer Programs for the Large Displacement Dynamic Response of Three Dimensional Solids," SAND76-0096, Sandia National Laboratories, Albuquerque, NM, 1976. (NNA.920811.0120)

[4] J. O. Hallquist, "User's manual for dyna2d—an explicit two-dimensional hydrodynamic finite element code with interactive rezoning," Lawrence Livermore National Laboratory, Livermore, California, February 1982. (NNA.920610.0028)

[5] J. O. Hallquist, "User's manuals for dyna3d and dynap," Lawrence Livermore National Laboratory, Livermore, California, July 1981. (NNA.920610.0029)

[6] L. M. Taylor and D. P. Flanagan, "PRONTO 2D: A Two Dimensional Transient Solid Dynamics Program," SAND86-0594, Sandia National Laboratories, Albuquerque, NM, March 1987. (NNA.900419.0008)

[7] L. M. Taylor and D. P. Flanagan, "PRONTO 3D: A Three Dimensional Transient Solid Dynamics Program," SAND87-1912, Sandia National Laboratories, Albuquerque, NM, March 1989. (NNA.920512.0111)

[8] B. Irons and A. Elsawaf, "The conjugate Newton algorithm for solving finite element equations," in *Formulations and Computation Algorithms in Finite Element Analysis* (K. J. Bathe and W. Wunderlich, eds.), MIT Press, 1977, pp. 655–672. (NNA.920512.0112)

[9] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 409–436, December 1952. (NNA.920626.0027)

[10] A. Ralston and H. S. Wilf, *Mathematical Methods of Digital Computers.* New York: John Wiley and Sons Inc., 1960, pp. 62–72. (NNA.920811.0121)

[11] P. Concus, G. H. Golub, and D. P. O'Leary, "A generalized conjugate gradient method for the numerical solution of elliptic partial differential equations," in *Sparse Matrix Computations* (J. R. Bunch and D. J. Rose, eds.), New York: Academic Press, 1976, pp. 309–332. (NNA.920512.0107)

[12] R. Fletcher and C. M. Reeves, "Function minimization by conjugate gradients," *The Computer Journal*, vol. 7, pp. 149–154, 1964. (NNA.920512.0113)

[13] M. L. Lenard, "Convergence conditions for restarted conjugate gradient methods with inaccurate line searches," *Mathematical Programming*, vol. 10, pp. 32–51, 1976. (NNA.920521.0051)

[14] M. J. D. Powell, "Restart procedures for the conjugate gradient method," *Mathematical Programming*, vol. 12, pp. 242–254, 1977. (NNA.920610.0030)

[15] J. W. Daniel, "The conjugate gradient method for linear and nonlinear operator equations," *SIAM Journal of Numerical Analysis*, vol. 4, no. 1, pp. 10–26, 1967. (NNA.920610.0031)

[16] R. Bartels and J. W. Daniel, "A conjugate gradient approach to nonlinear elliptic boundary value problems in irregular regions," in *Lecture Notes in Mathematics* (A. Dold and B. Eckmann, eds.), New York: Springer-Verlag, 1973, pp. 1–11. (NNA.920811.0122)

[17] G. D. Sjaardema, "Overview of the Sandia National Laboratories Engineering Analysis Code Access System," SAND92-2292, Sandia National Laboratories, Albuquerque, NM, January 1993. (NNA.930224.0105)

[18] L. M. Taylor, D. P. Flanagan, and W. C. Mills-Curran, "The GENESIS Finite Element Mesh File Format," SAND86-0910, Sandia National Laboratories, Albuquerque, NM, May 1986. (NNA.890906.0156)

[19] W. C. Mills-Curran, "EXODUS: A Finite Element File Format for Pre- and Post-Processing," SAND87-2997, Sandia National Laboratories, Albuquerque, NM, September 1988. (NNA.910813.0001)

[20] D. P. Flanagan and L. M. Taylor, "An accurate numerical algorithm for stress integration with finite rotations," *Computer Methods in Applied Mechanics and Engineering*, vol. 62, pp. 305–320, 1987. (NNA.920610.0032)

[21] J. K. Dienes, "On the analysis of rotation and stress rate in deforming bodies," *Acta Mechanica*, vol. 32, pp. 217–232, 1979. (NNA.900419.0007)

[22] G. C. Johnson and D. J. Bammann, "A discussion of stress rates in finite deformation bodies," *International Journal of Solids and Structures*, vol. 20, no. 8, pp. 725–737, 1984. (NNA.900417.0032)

[23] C. Truesdell, *A First Course in Rational Continuum Mechanics, Vol. 1, General Concepts.* New York: Academic Press, Inc., 1977, p. 162. (NNA.920512.0114)

[24] K. J. Bathe and E. L. Wilson, *Numerical Methods in Finite Element Analysis.* New Jersey: Prentice-Hall, 1976. (Not Cited)

[25] L. E. Malvern, *Introduction to the Mechanics of a Continuous Medium.* New Jersey: Prentice-Hall, Inc., 1969, pp. 226–228. (NNA.920512.0115)

[26] D. P. Flanagan and T. Belytschko, "A uniform strain hexahedron and quadrilateral with orthogonal hourglass control," *International Journal for Numerical Methods in Engineering*, vol. 17, pp. 679–706, 1981. (NNA.900319.0053)

[27] T. J. R. Hughes and J. Winget, "Finite rotation effects in numerical integration of rate constitutive equations arising in large-deformation analysis," *International Journal for Numerical Methods in Engineering*, vol. 15, no. 12, pp. 1862–1867, 1980. (NNA.920512.0108)

[28] R. D. Krieg and D. B. Krieg, "Accuracies of numerical solution methods for the elastic-perfectly plastic model," *ASME Journal of Pressure Vessel Technology*, vol. 99, pp. 510–515, 1977. (NNA.900417.0035)

[29] H. L. Schreyer, R. F. Kulak, and J. M. Kramer, "Accurate numerical solutions for elastic-plastic models," *ASME Journal of Pressure Vessel Technology*, vol. 101, pp. 226–234, 1979. (NNA.920521.0052)

[30] C. M. Stone, R. D. Krieg, and Z. E. Beisinger, "SANCHO—A Finite Element Computer Program for the Quasistatic, Large Deformation, Inelastic Response of Two-Dimensional Solids," SAND84-2618, Sandia National Laboratories, Albuquerque, NM, September 1988. (HQS.880517.2351)

[31] D. J. Benson and J. O. Hallquist, "A single surface contact algorithm for the post-buckling analysis of shell structures," *Computer Methods in Applied Mechanics and Engineering*, vol. 78, pp. 141–163, 1990. (NNA.920512.0109)

[32] J. T. Holden, "On the finite deflections of thin beams," *International Journal of Solids and Structures*, vol. 8, pp. 1051–1055, 1972. (HQS.880517.2302)

[33] A. Mendelson, *Plasticity: Theory and Application.* New York: The Macmillan Company, 1968, pp. 138–156. (NNA.920521.0053)

[34] J. Chakarabarty, *Theory of Plasticity.* New York: McGraw-Hill Book Company, 1987, pp. 306–315, 342–350. (NNA.930224.0104)

Blank Page

Appendix A
JAC3D v6.1 Input Instructions

Two input files are needed by the JAC3D program. Nodal point, element identification, boundary conditions, and contact surface data are supplied by a mesh generator in the GENESIS data format on unit 9. The second input file, unit 5, contains control information including problem definition, output options, solution control, boundary conditions, function data, and material properties. To associate sets of nodes or element sides with a boundary condition, flags are prescribed that correspond to the same flag for a set of either nodes or element sides in the GENESIS unit 9 file.

For the unit 5 file, free field input is used and each input line is identified by several descriptive words. The format is the description followed by data. The order of the input is not important except that data for each function and material must be grouped together. The first *three* letters of each word if it contains three letters must be spelled correctly and all words must be present. The free field input allows the user to delineate entries by either a blank, a comma, or an equal sign. A dollar sign indicates that whatever follows on the line of input is a comment. An asterisk indicates that the current input line is to be continued on the next line. An EXIT record will terminate the input. The following describes each line of data.

A.1 Problem Definition

TITLE

Enter a suitable title on the next line.

THERMAL PROBLEM *source, nfunct*

The default is an isothermal problem. If *source* is blank, then *nfunct* is ignored and temperatures must be supplied on unit 58 for a thermal problem. Temperature records are of the form

```
READ(58) TIME, (TEMP(I), I=1, NUMNP)
```

where NUMNP is the total number of nodes. If *source* is set to INTERNAL, a spatially-uniform temperature field is generated for each time step from the function *nfunct*, and unit 58 is not read.

DISTRIBUTED LOADS

The default is to apply no distributed body forces. Loads in force per unit mass must be supplied on unit 38 if this keyword is specified. Distributed load records are of the form

```
READ(36) TIME, ((DISTL(J,I), I=1, NUMNP), J=1,3)
```

where **NUMNP** is the total number of nodes.

LINEAR PROBLEM

The default is a geometrically nonlinear problem.

NONLINEAR PROBLEM

This is the default.

TRIAL LINEAR

Use of the trial linear option might help in accelerating solution convergence for problems involving large rotations. Two iteration passes will be attempted on the first load step. The first pass uses linear geometry assumptions and the second pass does a full nonlinear geometry solution using the linear solution as an initial guess. The default is that no trial linear solution will be calculated for the solution of the first load step.

INITIAL EQUILIBRIUM *etime*

This specifies that JAC calculate an initial equilibrium state before the first load step. *Etime* is a dummy time increment to be passed to any rate-dependent material models. Two equilibrium passes are made prior to beginning the load history; the displacements are zeroed out and the state variables reinitialized after each pass. Two passes are used because of geometrical nonlinearities; a stress state in equilibrium in a deformed configuration may not still be in equilibrium when displacements are rezeroed. The default is not to calculate initial equilibrium before taking the first load step.

RESTART PROBLEM

If a this command is input, a restart of the problem at the first time specified by the solution control function is executed. Restart data is expected on unit 32. All the input necessary to set up the problem must be included, i.e., GENESIS data should be on unit 9 and control information should be on unit 5. The restart file, unit 32, only contains the necessary information to establish initial mechanics conditions for the solution, and is written in the EXODUS file format. The default is not a restart problem.

A.2 Output Options

GENESIS OUTPUT

This specifies that the GENESIS mesh information be echoed on the ASCII output file, unit 6. GENESIS is the mesh portion of the EXODUS data format. The default is to not write detailed mesh information on the printed output.

ITERATION PRINT *n*

Several variables that will allow the monitoring of the iteration process during a

load step are written to the output file every n iterations. The default is for no intermediate information to be written.

OUTPUT FUNCTION *nfunct*

Nfunct specifies the function that will be used for determining when printed solution output will be written. The function $value(i)$ at $time(i)$ is the number of times the output will be written between $time(i)$ and $time(i+1)$. If this is not input, data for all the nodes and elements **will not be written.** If output for a load step is not selected, only an iteration summary is written to the output file, unit 6.

PLOT FUNCTION *nfunct*

Nfunct specifies the function that will be used for determining solution plot output times in the EXODUS format on unit 11. The function $value(i)$ at $time(i)$ is the number of times the output will be written between $time(i)$ and $time(i+1)$. If this is not input, data will be written for every time step.

WRITE RESTART FUNCTION *nfunct*

Nfunct specifies the function that will be used for determining the times when data is written to unit 30 for later use in restarting the problem. The function $value(i)$ at $time(i)$ is the number of times the output will be written between $time(i)$ and $time(i+1)$. If this command is not input, no restart data will be written.

A.3 Solution Control

SOLUTION FUNCTION *nfunct*

Nfunct specifies the function that will be used for determining solution time increments. The function $value(i)$ at $time(i)$ is the number of times the solution will be calculated between $time(i)$ and $time(i+1)$. Therefore,

$$\Delta time = \frac{time(i+1) - time(i)}{value(i)}$$

A SOLUTION FUNCTION command must be input.

DISPLACEMENT TOLERANCE *tolu*

This applies to the relative change in the norm of the incremental displacement vector within the last iteration. The default convergence tolerance is 1.0×10^{-12}.

RESIDUAL TOLERANCE *tolr*

This applies to the norm of the residual force vector divided by the norm of the applied loads vector. The default convergence tolerance is 1.0×10^{-3}.

MAXIMUM TOLERANCE *tolrmx*

This is a fallback tolerance which applies to the same quantity as *tolr*. Its default value is zero.

MAXIMUM RESIDUAL *residf*

This applies directly to the residual force vector. Convergence is assumed when its norm is less than *residf*. This criterion is used primarily in the absence of applied loads. *Residf* defaults to zero.

MAXIMUM ITERATIONS *nimax*

The default is the number of degrees of freedom.

If there are no significant applied loads, convergence is assumed if the norm of the residual force vector is less than *residf*. Otherwise, if either the displacement or residual tolerance is satisfied, the program assumes the load step has converged. If the convergence criteria are not satisfied within *nimax* iterations, JAC will first go back to the iterate $x_{j_{\min}}$, if any, that it was tracking for the smallest residual during the load step iterations (see the discussion under CGRESET LIMITS below). If the relative size of the corresponding residual $r_{j_{\min}}$ is less than *tolrmx*, then $x_{j_{\min}}$ is accepted and JAC will proceed to the next load step. If not, $x_{j_{\min}}$ is written to the plot file and the analysis is terminated.

CGRESET LIMITS *itstrt, itrset, tolfac*

These parameters control logic that resets the conjugate gradient iteration scheme. When the CG iteration is not converging, JAC picks a new initial guess "on the fly," selecting as its new guess the vector $x_{j_{\min}}$ which has produced the smallest residual $r_{j_{\min}}$ so far in the current load step. The first parameter, *itstrt*, specifies how many iterations to wait before looking for a minimum residual, i.e., JAC requires that $j_{\min} >$ *itstrt*. The default value is NDOF/100. *Itrset* specifies the number of iterations to allow between finding a minimum and restarting the CG algorithm. The default value of *itrset* is NDOF/2. The third parameter, *tolfac*, defines how much growth in the residual norm indicates divergence. Its default is 1000.

TRIAL VELOCITY FUNCTION *nfunct*

Nfunct specifies a function which will be used for determining a multiplier, which when applied to the last incremental displacement field, will generate an initial guess for this load step. The *value*(i) at *time*(i) will be used as the multiplier for all load steps between *time*(i) and *time*$(i+1)$ (values are not interpolated). For problems where the solution varies smoothly over many load steps, a trial velocity factor of 1.0 can significantly increase the rate of convergence. On the other hand, for problems with several disjoint load steps or abrupt changes in loading direction, a trial velocity can hinder convergence. The default is no trial velocity.

HOURGLASS PARAMETER *value*

Reduced-integration elements such as those in JAC are subject to hourglass ("zero-energy" or "spurious deformation") modes. The current formulation uses Flanagan's orthogonal stiffness formulation to control these modes. *Value* is a multiplier on the stiffness calculated by this formulation; generally the default value of 1.0 is suitable; in some problems where bending modes are important, reducing *value* by an order of magnitude or two may give a more accurate answer.

A.4 Boundary Conditions

X DISPLACEMENT *iflag, nfunct, mvalue*
Y DISPLACEMENT *iflag, nfunct, mvalue*
Z DISPLACEMENT *iflag, nfunct, mvalue*
X FORCE *iflag, nfunct, mvalue*
Y FORCE *iflag, nfunct, mvalue*
Z FORCE *iflag, nfunct, mvalue*

For displacement and force boundary conditions, the value of *iflag* must correspond to a nodeset flag in the GENESIS data file. If *nfunct* is zero or blank, then a zero condition is applied; otherwise the value of the function at the end of the load step times *mvalue*, the multiplier value, is applied. Linear interpolation is used to determine the function value for all loading functions. (The default value of *mvalue* is **zero.**)

XYPLANE DISPLACEMENT *iflag, angle*

This keyword specifies that there be no displacement normal to a plane perpendicular to the xy-plane of the analysis, and is useful for modeling certain symmetry conditions. *Iflag* is a nodeset boundary flag number and *angle* is the orientation of the constraint plane's normal vector, measured counterclockwise from the x-axis in degrees.

X GRAVITY FUNCTION *nfunct*
Y GRAVITY FUNCTION *nfunct*
Z GRAVITY FUNCTION *nfunct*

For gravity body forces, the total load is specified by using the value $f(t)$ of the specified function at the end of the load step along with the density, which is specified in the material data. *Load* = *density* × *volume* × $f(t)$, with *volume* calculated by the JAC3D program.

PRESSURE *iflag, nfunct, pvalue*

For pressure boundary conditions, *iflag* must correspond to a sideset flag in the GENESIS data base. The magnitude of the applied pressure is determined by multiplying the value of the function at the end of the load step by *pvalue*. The default value of *pvalue* is **zero.**

USER PRESSURE *iflag, pvalue*

This input record directs the code to call **SUBROUTINE BNDUPR** once each load step to get a pressure distribution. There is a dummy routine to serve as a template in **$ACCESS/ACCESS/analysis/jac3d/bndupr.f** which defines the calling arguments. *Iflag* must correspond to a sideset flag in the GENESIS data base. *Pvalue* is passed to the user subroutine in the **PBC** array.

SLIDING SURFACE *mflag, sflag, coeff, dtolr, ftolr, stolr*
FIXED SURFACE *mflag, sflag, dtolr*

Contact surfaces are specified by associating the master surface flag *mflag* and the slave surface flag *sflag* to corresponding sideset flags in the GENESIS data base.

Coeff is the coefficient of friction. Its default is zero.

Dtolr is a displacement tolerance such that if the two surfaces are within the tolerance, they are assumed to be in contact (in other words, a capture distance). Its units are relative to the length of the master element surface. The default is 1.0×10^{-7}.

Ftolr is the value of force that must be exceeded for a SLIDING SURFACE to be allowed to separate once contact has been established. The default is 1.0×10^{30}. A FIXED SURFACE is never allowed to separate once contact is established.

Stolr specifies the penetration distance within which to capture a slave node, in units of master-surface length. It defaults to 0.1.

A.5 Function Data

FUNCTION *n*
time(1), value(1)
time(2), value(2)

. .
. .
. .

time(m), value(m)
END

Function data are input as pairs of data, with one pair per record. The data is terminated by an END record. Care should be taken, because different input quantities make use of function data differently. It is suggested that the functions be numbered consecutively from one, since the function number n is used as an index into the function data array.

A.6 General Material Data

The following data may be input for all materials. The general material data must be accompanied by parameters specific to the material models, which are outlined in the following section.

MATERIAL *id, itype*

The *id* should correspond to an element block ID in the GENESIS data file. *Itype*

refers to the material type, and is optional if the material model title, such as ISOTHERMAL ELASTIC PLASTIC, appears on the next line.

TEMPERATURE *value_1, value_2, ... , value_n*
THERMAL STRAIN *value_1, value_2, ... , value_n*

An arbitrary number of pairs of temperature versus strain may be input, and linear interpolation is used to calculate the thermal strain at a specified temperature. The problem is assumed strain free initially, and loading is prescribed by calculating increments of thermal strain for each load step. (An increment in thermal strain divided by the corresponding temperature increment gives the coefficient of thermal expansion for a material.)

DENSITY *value*

The default is zero.

BIRTH TIME *value*

The default is zero.

DEATH TIME *value*

The default is 1.0×10^{30}. One set of birth and death times applies to all the elements in the problem with the same material ID.

INITIAL STRESS *sigxx0, sigyy0, sigzz0, tauxy0, tauyz0, tauzx0*
STRESS GRADIENT *gradxx, gradyy, gradzz, gradxy, gradyz, gradzx*

All the elements of the material are initialized with the stress components computed by, for instance,

$$\sigma_{xx0} = sigxx0 + z \times gradxx\,,$$

where z for each element is calculated by averaging the z-coordinates of all its nodes. The STRESS GRADIENT input record may be omitted if a constant initial stress is desired. The default is to apply no initial stress.

A.7 Specific Material Data

One set of the following data must be included with a set of general material data for each material. Enough data must be specified to define a real material. All values are defaulted to zero.

Material Type 1: Isothermal Elastic Plastic

The formulation of this model is described in SAND81-0998, "JAC—A Two-Dimensional Finite Element Computer Program for the Non-Linear Quasistatic Response of Solids with the Conjugate Gradient Method," pp. 22–24.

YOUNGS MODULUS *value*

POISSONS RATIO *value*

YIELD STRESS *value*

The default of zero results in an elastic material.

HARDENING MODULUS *value*

BETA *value*

Ranges from 0.0 to 1.0. Zero means only kinematic hardening occurs, while 1.0 means only isotropic hardening occurs.

END

An END record terminates each set of material data.

Material Type 2: Temperature Dependent Elastic Plastic

The formulation of this model is described in SAND81-0998, "JAC—A Two-Dimensional Finite Element Computer Program for the Non-Linear Quasistatic Response of Solids with the Conjugate Gradient Method," pp. 24–25.

YOUNGS MODULUS *value_1, value_2, ... , value_n*

POISSONS RATIO *value_1, value_2, ... , value_n*

YIELD STRESS *value_1, value_2, ... , value_n*

HARDENING MODULUS *value_1, value_2, ... , value_n*

The values of the temperature-dependent material parameters are given at the n temperatures specified on the TEMPERATURE input record for this material block. Values are interpolated to the temperature of the element. If the element temperature goes outside the range of temperatures given, values are extrapolated from the nearest two input values.

BETA *value*

Ranges from 0.0 to 1.0. Zero means only kinematic hardening occurs, while 1.0 means only isotropic hardening occurs.

END

An END record terminates each set of material data.

Material Type 3: Temperature Dependent Secondary Creep

The original formulation of this model is described in SAND81-0998, "JAC—A Two-Dimensional Finite Element Computer Program for the Non-Linear Quasistatic Response of Solids with the Conjugate Gradient Method," pp. 25–27, and uses a semi-analytic integration developed by R. Krieg. The integration has since been replaced by a vectorized

forward Euler method developed by C. Stone, H. Morgan, and M. Blanford, and will be documented in a forthcoming SAND report.

YOUNGS MODULUS *value_1, value_2, ... , value_n*

POISSONS RATIO *value_1, value_2, ... , value_n*

CREEP CONSTANT *value*

STRESS EXPONENT *value*

THERMAL CONSTANT *value*

END
An END record terminates each set of material data.

CREEP CONSTANT, STRESS EXPONENT, and THERMAL CONSTANT ($= Q/R$) are constant with respect to temperature. If a THERMAL PROBLEM is not specified, then the material temperature is assumed to be the *value_1* given on the TEMPERATURE record for this material block, and the first value for YOUNGS MODULUS and POISSONS RATIO is used.

Material Type 4: Isothermal Soil and Crushable Foam

This model was developed by R. Kreig, SC-DR-72-0883, "A Simple Constitutive Description for Soils and Crushable Foams." The JAC implementation is described in SAND81-0998, "JAC—A Two-Dimensional Finite Element Computer Program for the Non-Linear Quasistatic Response of Solids with the Conjugate Gradient Method," pp. 27–28. Note that the model now uses effective stress rather than J_2 in the formulation of the deviatoric yield function, following the PRONTO/SANTOS formulation.

SHEAR MODULUS *value*
BULK MODULUS *value*

A0 *value*
A1 *value*
A2 *value*

PFRAC *value*
PMAX *value*

VOLUME STRAIN *value_1, value_2, ... , value_8*

PRESSURE *value_1, value_2, ... , value_8*

END
An END record terminates each set of material data.

Material Type 5: Orthotropic Elastic

This model is as yet undocumented, but should be fairly self-explanatory. Currently material directions must be aligned with the global axes.

YOUNGS MODULUS E_{xx}, E_{yy}, E_{zz}

POISSONS RATIO ν_{yx}, ν_{zx}, ν_{zy}

SHEAR MODULUS G_{xy}, G_{yz}, G_{zx}

END

An END record terminates each set of material data.

For a transversely isotropic material,

$$\begin{aligned} E_{xx} = E_{yy} &= E_s \\ \nu_{xy} = \nu_{yx} &= \nu_s \\ \nu_{zx} = \nu_{zy} &= \nu_{sn} \\ G_{xy} = \frac{E_s}{2(1+\nu_s)} &= G_s \\ G_{yz} = G_{zx} &= G_{sn} \end{aligned}$$

Material Type 6: Isothermal Elastic Truss

This model is described in SAND87-1305, "JAC3D—A Three-Dimensional Finite Element Computer Program for the Nonlinear Quasi-Static Response of Solids with the Conjugate-Gradient Method."

YOUNGS MODULUS *value*

AREA *value*

All the truss elements using this material ID have the same area.

END

An END record terminates each set of material data.

Material Type 7: Power Law Hardening Elastic Plastic

This model was developed by C. Stone, G. Wellman, and R. Kreig at Sandia Labs New Mexico. For documentation, see SAND90-0153, "A Vectorized Elastic/Plastic Power Law Hardening Material Model Including Lüders Strain."

YOUNGS MODULUS *value*

POISSONS RATIO *value*

YIELD STRESS *value*
The default of zero results in an elastic material.

HARDENING CONSTANT *value*

HARDENING EXPONENT *value*

LUDERS STRAIN *value*

END
An END record terminates each set of material data.

Material Type 10: Sandia Damage Model

This model was developed by D. Bammann, G. Johnson, and M. Chiesa at Sandia Labs California. For documentation, see for instance SAND90-8227, "A Strain Rate Dependent Flow Surface Model of Plasticity."

YOUNGS MODULUS *value_1, value_2, ... , value_n*

POISSONS RATIO *value_1, value_2, ... , value_n*
If a THERMAL PROBLEM is not specified, then the material temperature is assumed to be the *value_1* given on the TEMPERATURE record for this material block, and the first value for YOUNGS MODULUS and POISSONS RATIO is used.

HEAT COEFFICIENT *value*
Heat generated from plastic work ($\frac{0.9}{\rho C_v}$). Not currently implemented.

INITIAL DAMAGE *value*
Must be nonzero for evolution of damage to occur.

DAMAGE CONSTANT *value*

C1 *value*
C2 *value*
$V = c_1 e^{-c_2/\theta}$

C3 *value*
C4 *value*
$Y = c_3 e^{c_4/\theta}$

C5 *value*
C6 *value*
$f = c_5 e^{-c_6/\theta}$

C7 *value*
C8 *value*

$r_d = c_7 e^{-c_8/\theta}$

C9 *value*
C10 *value*

$h = c_9 e^{c_{10}/\theta}$

C11 *value*
C12 *value*

$r_s = c_{11} e^{-c_{12}/\theta}$

C13 *value*
C14 *value*

$R_d = c_{13} e^{-c_{14}/\theta}$

C15 *value*
C16 *value*

$H = c_{15} e^{c_{16}/\theta}$

C17 *value*
C18 *value*

$R_s = c_{17} e^{-c_{18}/\theta}$

END

An END record terminates each set of material data.

Appendix B
JAC3D v6.1 Output Description

B.1 General Printed Output

Printed output begins by echoing the input data from unit 5. The GENESIS input data is also echoed if the GENESIS OUTPUT input record has been specified. A large amount of data should be expected when printing the GENESIS data. The input section ends with summary of the dynamic memory allocation for the problem.

Each load step produces at least one page of printed output. The first page is printed even if the user has not requested output for the time step.

```
SOLUTION TIME =  time

OUTPUT FOR LOAD INCREMENT     n

SUM OF X REACTIONS ------------------   rx
SUM OF Y REACTIONS ------------------   ry
SUM OF Z REACTIONS ------------------   rz
NORM OF APPLIED FORCES --------------   fn
RESIDUAL FORCE NORM -----------------   rn
RESIDUAL FORCE TOLERANCE ------------   rnn
INCREMENTAL DISPLACEMENT NORM --------   un
INCREMENTAL DISPLACEMENT TOLERANCE ---   unn
CPU TIME FOR LOAD STEP --------------   t1
TOTAL CPU TIME ----------------------   t2
NUMBER OF ITERATIONS ON FIRST PASS ----  npass(1)
NUMBER OF ITERATIONS ON SECOND PASS ---  npass(2)
TRIAL VELOCITY FACTOR ---------------   factd
```

Rx, ry, and rz are the total reactions resulting from application of the loads. They can be compared to the applied loads to obtain a measure of the state of convergence of the solution. Of particular interest are the reactions in directions where the applied load is zero. The conjugate gradient method will generally get the reaction in the direction of the nonzero applied load first in the iteration process, and then reduce the other reactions to zero.

The applied force norm **fn** and displacement norm **un** are used to calculate **rnn** (= **rn/fn**) and **unn**, respectively, which are compared to the specified residual tolerance *tolr* and displacement tolerance *tolu*. The residual force norm **rn** is also compared directly to the specified maximum residual *residf*. These quantities can be used as a measure of how close the load step has come in the event the program stops before satisfying convergence.

The amount of central processor computer time is given for the current load step and the total time used for the job.

The number of iterations that are used for the current load step is printed to give the user a measure of how fast the load step has converged. If either the TRIAL LINEAR or LINEAR PROBLEM option is used, **npass(1)** is the number of iterations of convergence of the trial linear or linear solution. For the TRIAL LINEAR option, **npass(2)** is the number of iterations needed for the nonlinear solution to the first load step. For a nonlinear problem without the TRIAL LINEAR option, **npass(1)** is the total number of iterations needed for the load step.

Factd is a multiplier used to scale the velocity vector from the previous load step to obtain a trial velocity vector as an initial guess for the current load step.

Further output is produced at the solution times specified by the OUTPUT FUNCTION. Current coordinates and displacements of each node are listed, with **X**, **Y**, and **Z** denoting the coordinates and **DISX**, **DISY**, and **DISZ** denoting the components of total displacement. Components of stress in the current deformed configuration carry the labels **SIGX**, **SIGY**, **SIGZ**, **SIGXY**, **SIGYZ**, and **SIGZX**. The element temperature is called **TEMP**. Strains are not calculated or output by the program. If a load step does not converge, current values of these variables are written to the output file for diagnostic purposes before the program exits.

B.2 Contact Surface Printed Output

Contact surface data is printed in the following format at the times specified by the OUTPUT FUNCTION.

```
INTERFACE NUMBER      n

NUMBER  MASTER   SLAVE        LOCATION ON ELEMENT      NORMAL      NORMAL
        ELEMENT  NODE           ZETA         ETA    DISPLACEMENT  FORCE
```

The output is ordered first by **INTERFACE NUMBER**, and then by the **NUMBER** in the overall list of interface slave nodes.

MASTER ELEMENT is the number in a list of the total number of master element faces with which the **SLAVE NODE** is interacting. **SLAVE NODE** lists the node's global node number. An interface condition of **FIXED**, **SLIDING**, or **NO CONTACT** is indicated beside the node number. For a slave node to be in contact, the values of **ZETA** and **ETA** must be greater than -1.001 and less than $+1.001$. A slave node which is penetrating its master element face will have a **NORMAL DISPLACEMENT** less than the *dtolr* value specified for the contact surface. For a sliding surface slave node to be in contact with a master element face, the value of the **NORMAL FORCE** must be less than the value of *ftolr* specified for the surface.

B.3 Interim Iteration Printed Output

If the ITERATION PRINT command is used, the following output is obtained every n iterations.

```
LOAD  NUMBER OF      X          Y          Z        FORCE    DISPLACEMENT
STEP  ITERATIONS  REACTION   REACTION   REACTION  TOLERANCE   TOLERANCE
```

The quantities listed under the last five headings are the intermediate values of **rx**, **ry**, **rz**, **rnn**, and **unn**, respectively, which are described in Section B.1.

B.4 Plot Data Output

A plot output file is written in the EXODUS format on unit 11 by JAC3D for post-processing. The following variables are written to the EXODUS file for each solution time specified by the PLOT FUNCTION. If a load step does not converge, current values of these variables are written to the EXODUS file for diagnostic purposes before the program exits.

B.4.1 Global Variables

Name	Description
RN	Residual force norm
RNN	Residual force tolerance
UN	Incremental displacement norm
UNN	Incremental displacement tolerance
FN	Applied force norm
RX	Total reaction force in the x-direction
RY	Total reaction force in the y-direction
RZ	Total reaction force in the z-direction
ITER	Cumulative total number of iterations

B.4.2 Nodal Variables

Name	Description
DISPLX	Total displacement in the x-direction
DISPLY	Total displacement in the y-direction
DISPLZ	Total displacement in the z-direction
RESIDX	Total residual force in the x-direction
RESIDY	Total residual force in the y-direction
RESIDZ	Total residual force in the z-direction

Components of the residual forces—RESIDX, RESIDY, and RESIDZ—are the values of the residuals at the end of the load step. If the load step does not converge, it is sometimes useful to plot the residuals to identify areas of the mesh where convergence is difficult.

B.4.3 Element Variables

Continuum Element

Name	Description
SIGXX	Stress σ_{xx}
SIGYY	Stress σ_{yy}
SIGZZ	Stress σ_{zz}
TAUXY	Stress τ_{xy}
TAUYZ	Stress τ_{yz}
TAUZX	Stress τ_{zx}
EPX1 EPX2 ⋮	Material history variables
TEMP	Temperature at the center of the element

Truss Element

Name	Description
SIGXX	Axial stress
TEMP	Temperature at the center of the element

B.4.4 Material History Variable Names

The values in the history variable arrays depend upon the material models used in the analysis. The following is a description of the variables by material type.

Material Type 1: Isothermal Elastic Plastic

Name	Description
EPX1	Yield surface center stress component σ_{xx}
EPX2	Yield surface center stress component σ_{yy}
EPX3	Yield surface center stress component σ_{zz}
EPX4	Yield surface center stress component τ_{xy}
EPX5	Yield surface center stress component τ_{yz}
EPX6	Yield surface center stress component τ_{zx}
EPX7	Accumulated plastic strain

Material Type 2: Temperature Dependent Elastic Plastic

Name	Description
EPX1	Yield surface center stress component σ_{xx}
EPX2	Yield surface center stress component σ_{yy}
EPX3	Yield surface center stress component σ_{zz}
EPX4	Yield surface center stress component τ_{xy}
EPX5	Yield surface center stress component τ_{yz}
EPX6	Yield surface center stress component τ_{zx}
EPX7	Yield surface radius

Material Type 3: Temperature Dependent Secondary Creep

Name	Description
EPX1	Accumulated creep strain

Material Type 4: Isothermal Soil and Crushable Foam

Name	Description
EPX1	Maximum previous volumetric strain (positive in compression)
EPX2	Current value of volumetric strain at which tensile fracture will occur
EPX3	Current value of volumetric strain (positive in compression)

Material Type 5: Orthotropic Elastic

There are no history variables for this material model.

Material Type 6: Isothermal Elastic Truss

There are no history variables for this material model.

Material Type 7: Power Law Hardening Elastic Plastic

Name	Description
EPX1	Accumulated plastic strain

Material Type 10: Sandia Damage Model

Name	Description
EPX1	Backstress component α_{xx}
EPX2	Backstress component α_{yy}
EPX3	Backstress component α_{zz}
EPX4	Backstress component α_{xy}
EPX5	Backstress component α_{yz}
EPX6	Backstress component α_{zx}
EPX7	Hardening scalar k
EPX8	Void fraction ϕ
EPX9	Void fraction rate $\dot{\phi}$

Appendix C
Adding a New Constitutive Model to JAC3D

A material model is identified by a name and a number. For example, the isothermal elastic/plastic model is named ISOTHERMAL ELASTIC PLASTIC and is Material Type 1. A new material model can easily be added to the JAC3D code. The interface consists of the following items.

C.1 Initialization of Variables

Several COMMON variables associated with the material models are initialized in the main program. MAXMC is the second dimension of the EE array used to store temperature/ thermal-strain pairs and material constants. The leading dimension of this array may dynamically grow to the maximum number of temperature points specified on a TEMPERATURE input record, but will always be at least 8. Therefore, there are $8 \times (\texttt{MAXMC} - 2)$ locations available to the material model to store material constants. If more space than this is needed, MACMC may be increased. MAXMCT defines the maximum number of material constants for any material that has a temperature dependence, and is used to dimension the CM array. Thus if the program contained only the isothermal elastic/plastic model, these variables would be initialized as shown below.

```
MAXMC = 7
MAXMCT = 0
```

Associated with the material number is the number of state variables in the model (other than stress). In the main program, the following assignments specify the number of state variables for each model.

```
IEPSIZ(1) = 7
IEPSIZ(2) = 7
IEPSIZ(3) = 1
IEPSIZ(4) = 3
IEPSIZ(5) = 0
IEPSIZ(6) = 0
IEPSIZ(7) = 1
IEPSIZ(8) = 0
IEPSIZ(9) = 0
IEPSIZ(10) = 9
```

For example, the isothermal elastic/plastic model (Material Type 1) needs seven storage locations (six backstress components and one storage location for the effective plastic strain). As presently programmed, ten models can be included. The number of models can be increased by changing the dimension of the IEPSIZ array in the EPSZ common block:

```
      COMMON EPSZ/IEPSIZ(10),NEPSIZ/
```

If the maximum number of state variables is increased beyond 12 for any material model, then additional names must be added to the arrays NAMEEL and NMRSEL in SUBROUTINE IN, and the PARAMETERs that govern their size must be increased correspondingly. By convention, the state variable names begin with EPX.

```
      PARAMETER (NCORD=3, NVARNP=9, NVAREL=21, NGLOBL=9)
      PARAMETER (NVRSNP=6, NVRSEL=46)
      DATA NAMEEL/'SIGXX','SIGYY','SIGZZ','TAUXY',
     1   'TAUYZ','TAUZX','SIGT','TEMP','DEATH',
     2   'EPX1','EPX2','EPX3','EPX4','EPX5','EPX6',
     3   'EPX7','EPX8','EPX9','EPX10','EPX11','EPX12'/
      DATA NMRSEL/'SIGXX','SIGYY','SIGZZ',
     1            'TAUXY','TAUYZ','TAUZX','SIGT',
     2   'EPX1','EPX2','EPX3','EPX4', 'EPX5', 'EPX6',
     3   'EPX7','EPX8','EPX9','EPX10','EPX11','EPX12',
     4   'R11','R21','R31','R12','R22','R32','R13','R23','R33',
     5   'STRECHXX','STRECHYY','STRECHZZ',
     6   'STRECHXY','STRECHYZ','STRECHZX',
     7   'HG1X','HG1Y','HG1Z','HG2X','HG2Y','HG2Z',
     8   'HG3X','HG3Y','HG3Z','HG4X','HG4Y','HG4Z'/
```

Here NVAREL is the dimension of the NAMEEL array, while NVRSEL is the dimension of the NMRSEL array.

C.2 Material Input and Output

The procedure for reading and echoing model data will now be described. The isothermal elastic/plastic model will be used as an example of the statements that must be included for a new material. First, in SUBROUTINE IN, the material type number is stored in the array MTYPE.

```
          ELSE IF(WORD1 .EQ. 'ISO' .AND. WORD2 .EQ. 'ELA' .AND.
     1       WORD3 .EQ. 'PLA') THEN
             MTYPE(MATNUM) = 1
```

The material constants to be input are stored in array EE(NTMAX,MAXMC,*), where NTMAX is at least 8. An arbitrary number of values for each material constant may be used to describe the material's dependence on temperature. A maximum of three words may be used to name a material constant.

```
            IF(MATTYPE .EQ. 1) THEN
C
C     ISOTHERMAL ELASTIC PLASTIC MATERIAL
C
                IF( WORD1 .EQ. 'YOU' .AND. WORD2 .EQ. 'MOD') THEN
                   EE(1,3,MATNUM) = RVALUE(3)
                ELSE IF( WORD1 .EQ. 'POI' .AND. WORD2 .EQ. 'RAT') THEN
                   EE(1,4,MATNUM) = RVALUE(3)
                ELSE IF( WORD1 .EQ. 'YIE' .AND. WORD2 .EQ. 'STR') THEN
                   EE(1,5,MATNUM) = RVALUE(3)
                ELSE IF( WORD1 .EQ. 'HAR' .AND. WORD2 .EQ. 'MOD') THEN
                   EE(1,6,MATNUM) = RVALUE(3)
                ELSE IF( WORD1 .EQ. 'HAR' .AND. WORD2 .EQ. 'PAR') THEN
                   EE(1,7,MATNUM) = RVALUE(3)
                ELSE IF( WORD1 .EQ. 'BET' ) THEN
                   EE(1,7,MATNUM) = RVALUE(3)
                END IF
```

Here the values are stored beginning with 3 as the second array index. Locations 1 and 2 are used to store the temperatures and thermal strains, respectively. If a material parameter had a temperature dependence, its values would be stored by varying the first index of the EE array from 1 to NTMAX, the maximum number of temperature points input. This model allows the keyword HARDENING PARAMETER as a synonym for BETA.

The material constants are echoed in SUBROUTINE PRINTMT. As an example of the statements that are needed, the following statements are used for the isothermal elastic/plastic model.

```
IF(MTYPE(N).EQ.1) WORD = 'ISOTHERMAL ELASTIC PLASTIC'
    .
    .
    .
IF(MTYPE(N) .EQ. 1) THEN
   WRITE (KOUT,5050)  EE(1,3,N)
   WRITE (KOUT,5060)  EE(1,4,N)
   WRITE (KOUT,5070)  EE(1,5,N)
   WRITE (KOUT,5080)  EE(1,6,N)
   WRITE (KOUT,5090)  EE(1,7,N)
```

The following FORMAT statements are used for the isothermal elastic/plastic model.

```
5050 FORMAT(' YOUNGS MODULUS      ',1X,G10.3)
5060 FORMAT(' POISSONS RATIO      ',1X,G10.3)
5070 FORMAT(' YIELD STRESS        ',1X,G10.3)
5080 FORMAT(' HARDENING MODULUS   ',1X,G10.3)
5090 FORMAT(' BETA                ',1X,G10.3)
```

C.3 Calling the Material Model

The call to the material model which updates the stress every CG iteration must be added to SUBROUTINE RESFORH. By convention, the name of the material model subroutine is SE (for σ-ϵ, stress-strain) followed by the material type number. The following statement calls the isothermal elastic/plastic model.

```
                IF(MTYPE(IMM).EQ.1) CALL SE1(EE(1,1,IMM),
     1             CH(1,LFT),TSIG(1,LFT),TEPX(NEP),DVEL,
     2             DTHS(LFT),NEL)
```

Here EE contains the material constants and CH returns the 21 entries defining a modulus to be used for the CG line search. TSIG is the current stress in the deformed but unrotated configuration, TEPX stores the current state variables, DVEL contains the velocity gradients, DTHS contains the thermal strains for thermally-loaded problems, and NEL is the number of elements in the current vector block to be processed by the constitutive model.

C.4 Interpolating Temperature-Dependent Material Constants

If the new model has moduli that vary with temperature, then SUBROUTINE INTERP1 must be modified. The interpolated material properties are stored in array CM. Values for the material constants at the beginning and the end of the time step must be calculated. As an example, the following interpolates four material constants at the beginning and the end of the time step.

```
C
C     INTERPOLATE AT THE BEGINNING OF THE STEP
C
                .
                .
                .
          RATIO = (TH1(I)-EE(M-1,1))/(EE(M,1)-EE(M-1,1))
```

```
      IF(MTYPE.EQ.2) THEN
         CM(1,1,I) = EE(M-1,3) + RATIO*(EE(M,3)-EE(M-1,3))
         CM(1,2,I) = EE(M-1,4) + RATIO*(EE(M,4)-EE(M-1,4))
         CM(1,3,I) = EE(M-1,5) + RATIO*(EE(M,5)-EE(M-1,5))
         CM(1,4,I) = EE(M-1,6) + RATIO*(EE(M,6)-EE(M-1,6))
      END IF
         .
         .
         .
C
C     INTERPOLATE AT THE END OF THE STEP
C
         .
         .
         .
      RATIO = (TH2(I)-EE(M-1,1))/(EE(M,1)-EE(M-1,1))
      IF(MTYPE.EQ.2) THEN
         CM(2,1,I) = EE(M-1,3) + RATIO*(EE(M,3)-EE(M-1,3))
         CM(2,2,I) = EE(M-1,4) + RATIO*(EE(M,4)-EE(M-1,4))
         CM(2,3,I) = EE(M-1,5) + RATIO*(EE(M,5)-EE(M-1,5))
         CM(2,4,I) = EE(M-1,6) + RATIO*(EE(M,6)-EE(M-1,6))
      END IF
```

C.5 The Diagonal of the Stiffness Matrix

SUBROUTINE DIAG calculates the preconditioning matrix M to be the diagional of the linear stiffness matrix, as described in Section 3.3. The two quantities $(\lambda+2\mu)/V$ and μ/V must be supplied for each element in the material block, where λ and μ are the Lamé parameters for the material and V is the element volume. For example, the following code is for the isothermal elastic/plastic material model. For this model, EE(1,3) contains Young's modulus and EE(1,4) contains Poisson's ratio.

```
      IF(MTYPE.EQ.1) THEN
         E1 = EE(1,3)*(1.0-EE(1,4))/((1.0+EE(1,4))*(1.0-2.0*EE(1,4)))
         E2 = 0.5*EE(1,3)/(1.0+EE(1,4))
         DO 100 I = 1,NEL
            C(1,I) = E1/VOL(I)
            C(2,I) = E2/VOL(I)
  100    CONTINUE
```

If the new model has temperature-dependent material constants, then the interpolated values in the CM array should be used instead of the values in the EE array.

C.6 The Effective Bulk Modulus

SUBROUTINE THFOR uses an effective bulk modulus to calculate the thermal expansion contribution to the applied loads vector. (The applied loads vector is used only to scale the residual force norm for comparison with the residual tolerance.) The elastic bulk modulus must be supplied for each element in the material block. The following code performs this calculation for the isothermal elastic/plastic model, where EE(1,3) contains Young's modulus and EE(1,4) contains Poisson's ratio.

```
      IF(MTYPE.EQ.1) THEN
         BULK = EE(1,3)/(1.0-2.0*EE(1,4))
         DO 100 I = 1,NEL
            THSIG(I) = BULK*DTHS(I)
100      CONTINUE
```

As in SUBROUTINE DIAG, if the model has temperature-dependent material constants, CM should be used instead of EE.

C.7 The Effective Shear Modulus

SUBROUTINE CVIS calculates an effective shear modulus for use in the hourglass control logic, as described in Section 4.1.1. The elastic shear modulus must be supplied for each element in the material block. The following code performs this calculation for the isothermal elastic/plastic model, where EE(1,3) contains Young's modulus and EE(1,4) contains Poisson's ratio.

```
      IF(MTYPE.EQ.1) THEN
         DO 100 I = 1,NEL
            GA(I) = 0.5*EE(1,3)/(1.0+EE(1,4))
100      CONTINUE
```

As in SUBROUTINE DIAG, if the model has temperature-dependent material constants, CM should be used instead of EE.

C.8 The Line Search Modulus

The moduli needed to perform the CG line search as described in Section 3.4.2 are returned by the material model subroutine in the array C. Twenty-one entries in the (assumed symmetric) 6×6 modulus matrix relating the components of the current stress increment to the current strain increment must be supplied by the model, stored as follows:

```
C(1)   C(2)   C(6)   C(8)   C(11)  C(16)
C(2)   C(3)   C(7)   C(9)   C(12)  C(17)
C(6)   C(7)   C(5)   C(10)  C(13)  C(18)
C(8)   C(9)   C(10)  C(4)   C(14)  C(19)
C(11)  C(12)  C(13)  C(14)  C(15)  C(20)
C(16)  C(17)  C(18)  C(19)  C(20)  C(21)
```

For the isothermal elastic/plastic model, the secant modulus derived in Section 5.1.6 is used. If a secant modulus is not available, an elastic modulus or a tangent modulus may be used. However, the convergence rate of the CG algorithm is affected by the accuracy of the line search.

C.9 Example of a Material Model Subroutine: Isothermal Elastic/Plastic

```
      SUBROUTINE SE1(EE,C,SIG,EPX,DVEL,DTHS,NEL)
C=======================================================================
C
C     ELASTIC-PLASTIC MATERIAL WITH ISOTROPIC AND KINEMATIC HARDENING
C     THE ROUTINE COMPUTES THE STRESSES FOR NEL ELEMENTS FOR THE
C     JAC3D PROGRAM.
C
C
C     DVEL CONTAINS INCREMENTAL STRAINS WITH DXY, DYZ, AND DZX BEING
C     ENGINEERING SHEAR STRAIN QUANTITIES
C
C     SEVEN HISTORY VARIABLE OR PLOT VARIABLES FOR THE CONSTITUTIVE
C     MODEL ARE STORED IN ARRAY EPX
C
C     THE CONSTITUTIVE ARRAY C CONTAINS THE ELASTIC OR SECANT
C     CONSTITUTIVE MATRIX STORED AS
C
C        C(1)    C(2)    C(6)    C(8)    C(11)   C(16)
C
C        C(2)    C(3)    C(7)    C(9)    C(12)   C(17)
C
C        C(6)    C(7)    C(5)    C(10)   C(13)   C(18)
C
C        C(8)    C(9)    C(10)   C(4)    C(14)   C(19)
C
C        C(11)   C(12)   C(13)   C(14)   C(15)   C(20)
C
C        C(16)   C(17)   C(18)   C(19)   C(20)   C(21)
C
C     THE THERMAL STRAIN IS STORED IN THE DTHS ARRAY
C
      INCLUDE 'COMIN'
C
      DIMENSION EE(NTMAX,*),C(21,*),SIG(6,*),EPX(7,*),DVEL(9,*),DTHS(*)
C
      THIRD = 1.0/3.0
      QH = EE(1,3)*EE(1,6)/(EE(1,3)-EE(1,6))
      QB = EE(1,7)
      QS = EE(1,5)
      C1 = EE(1,3)*EE(1,4)/((1.0+EE(1,4))*(1.0-2.0*EE(1,4)))
```

```
      G2 = EE(1,3)/(1.0+EE(1,4))
      G = 0.5*G2
      QBQH = QB*QH
C
C     COMPUTE TRIAL STRESS
C
      DO 100 I = 1,NEL
         DXX = DVEL(1,I)
         DYY = DVEL(2,I)
         DZZ = DVEL(3,I)
         DXY = 2.0*DVEL(4,I)
         DYZ = 2.0*DVEL(5,I)
         DZX = 2.0*DVEL(6,I)
         P = C1*(DXX + DYY + DZZ-3.0*DTHS(I))
         SIG(1,I) = SIG(1,I)+P+G2*(DXX-DTHS(I))
         SIG(2,I) = SIG(2,I)+P+G2*(DYY-DTHS(I))
         SIG(3,I) = SIG(3,I)+P+G2*(DZZ-DTHS(I))
         SIG(4,I) = SIG(4,I)+G*DXY
         SIG(5,I) = SIG(5,I)+G*DYZ
         SIG(6,I) = SIG(6,I)+G*DZX
  100 CONTINUE
C
      A11 = G2*THIRD
      A33 = A11 + C1
      IF(QS.EQ.0.0) THEN
         A332A1 = A33+2.0*A11
         A33A1 = A33-A11
         A115 = 1.5*A11
         DO 110 I = 1,NEL
            C(1,I) = A332A1
            C(3,I) = A332A1
            C(5,I) = A332A1
            C(2,I) = A33A1
            C(6,I) = A33A1
            C(7,I) = A33A1
            C(4,I) = A115
            C(15,I) = A115
            C(21,I) = A115
            C(8,I) = 0.0
            C(9,I) = 0.0
            C(10,I) = 0.0
            C(11,I) = 0.0
            C(12,I) = 0.0
```

```
            C(13,I) = 0.0
            C(14,I) = 0.0
            C(16,I) = 0.0
            C(17,I) = 0.0
            C(18,I) = 0.0
            C(19,I) = 0.0
            C(20,I) = 0.0
  110     CONTINUE
        ELSE
C
          FAC1 = 1.0/(1.5*G2+QH)
          FAC2 = 1.5*G2
          FAC3 = (1.0-QB)*QH
          A11 = G2*THIRD
          A22 = 2.25*G2*G2*FAC1
          A33 = A11 + C1
C
          DO 120 I = 1,NEL
             AK = QS+QBQH*EPX(7,I)
             DA1 = SIG(1,I)-EPX(1,I)
             DA2 = SIG(2,I)-EPX(2,I)
             DA3 = SIG(3,I)-EPX(3,I)
C
             Q1 = (2.0*DA1-DA2-DA3)*THIRD
             Q2 = (2.0*DA2-DA1-DA3)*THIRD
             Q3 = (2.0*DA3-DA1-DA2)*THIRD
             Q4 = SIG(4,I)-EPX(4,I)
             Q5 = SIG(5,I)-EPX(5,I)
             Q6 = SIG(6,I)-EPX(6,I)
C
             AJ2 = Q4*Q4-Q1*Q2-Q2*Q3-Q1*Q3+Q5*Q5+Q6*Q6
             AK2 = 3.0*AJ2-AK*AK
C
             SCLE = 0.5*(1.0+SIGN(1.0,AK2))
C
             AJ1 = SQRT(3.0*ABS(AJ2))+1.0-SCLE
C
             DEPI = SCLE*FAC1*(AJ1-AK)
             EPX(7,I) = EPX(7,I)+DEPI
             DEPS = SCLE*FAC2*DEPI/AJ1
             DEPN = SCLE*FAC3*DEPI/AJ1
C
             SIG(1,I) = SIG(1,I)-DEPS*Q1
```

```
                SIG(2,I) = SIG(2,I)-DEPS*Q2
                SIG(3,I) = SIG(3,I)-DEPS*Q3
                SIG(4,I) = SIG(4,I)-DEPS*Q4
                SIG(5,I) = SIG(5,I)-DEPS*Q5
                SIG(6,I) = SIG(6,I)-DEPS*Q6
C
                EPX(1,I) = EPX(1,I)+DEPN*Q1
                EPX(2,I) = EPX(2,I)+DEPN*Q2
                EPX(3,I) = EPX(3,I)+DEPN*Q3
                EPX(4,I) = EPX(4,I)+DEPN*Q4
                EPX(5,I) = EPX(5,I)+DEPN*Q5
                EPX(6,I) = EPX(6,I)+DEPN*Q6
C
                A1 = A11*(1.0-DEPS*SCLE)
                A2 = A22*AK*SCLE/(AJ1**3.0)
C
                C(1,I) = A33 + 2.0*A1 - A2*Q1*Q1
                C(3,I) = A33 + 2.0*A1 - A2*Q2*Q2
                C(5,I) = A33 + 2.0*A1 - A2*Q3*Q3
                C(2,I) = A33 - A1 - A2*Q2*Q1
                C(6,I) = A33 - A1 - A2*Q1*Q3
                C(7,I) = A33 - A1 - A2*Q2*Q3
                C(4,I) =   1.5*A1 - A2*Q4*Q4*0.5
                C(15,I) =  1.5*A1 - A2*Q5*Q5*0.5
                C(21,I) =  1.5*A1 - A2*Q6*Q6*0.5
                C(8,I) =  -A2*Q1*Q4
                C(9,I) =  -A2*Q2*Q4
                C(10,I) = -A2*Q3*Q4
                C(11,I) = -A2*Q5*Q1
                C(12,I) = -A2*Q5*Q2
                C(13,I) = -A2*Q5*Q3
                C(14,I) = -A2*Q5*Q4
                C(16,I) = -A2*Q6*Q1
                C(17,I) = -A2*Q6*Q2
                C(18,I) = -A2*Q6*Q3
                C(19,I) = -A2*Q6*Q4
                C(20,I) = -A2*Q6*Q5
  120     CONTINUE
C
        END IF
        RETURN
        END
```

Appendix D
RIB and SEPDB Information

The following information is provided for the benefit of the Yucca Mountain Project.

D.1 Information from the Reference Information Base Used in this Report

This report contains no information from the Reference Information Base.

D.2 Candidate Information for the Reference Information Base

This report contains no candidate information for the Reference Information Base.

D.3 Candidate Information for the Site and Engineering Properties Data Base

This report contains no candidate information for the Site and Engineering Properties Data Base.

Blank Page

YUCCA MOUNTAIN SITE CHARACTERIZATION PROJECT

DISTRIBUTION LIST

1 J. W. Barlett (RW-1)
Director
OCRWM
US Department of Energy
1000 Independence Avenue SW
Washington, DC 20585

1 F. G. Peters (RW-2)
Deputy Director
OCRWM
US Department of Energy
1000 Independence Avenue SW
Washington, DC 20585

1 T. H. Isaacs (RW-4)
Office of Strategic Planning and International Programs
OCRWM
US Department of Energy
1000 Independence Avenue SW
Washington, DC 20585

1 J. D. Saltzman (RW-5)
Office of External Relations
OCRWM
US Department of Energy
1000 Independence Avenue SW
Washington, DC 20585

1 Samuel Rousso (RW-10)
Office of Program and Resource Mgt.
OCRWM
US Department of Energy
1000 Independence Avenue SW
Washington, DC 20585

1 J. C. Bresee (RW-10)
OCRWM
US Department of Energy
1000 Independence Avenue SW
Washington, DC 20585

1 C. P. Gertz (RW-20)
Office of Geologic Disposal
OCRWM
US Department of Energy
1000 Independence Avenue, SW
Washington, DC 20585

1 S. J. Brocoum (RW-22)
Analysis and Verification Division
OCRWM
US Department of Energy
1000 Independence Avenue SW
Washington, DC 2585

1 J. Roberts, Acting Associate Director (RW-30)
Office of Systems and Compliance
OCRWM
US Department of Energy
1000 Independence Avenue, SW
Washington, DC 20585

1 J. Roberts (RW-33)
Director, Regulatory Compliance Division
OCRWM
US Department of Energy
1000 Independence Avenue, SW
Washington, DC 20585

1 G. J. Parker (RW-332)
OCRWM
US Department of Energy
1000 Independence Avenue, SW
Washington, DC 20585

1 R. A. Milner (RW-40)
Office of Storage and Transporation
OCRWM
US Department of Energy
1000 Independence Avenue, SW
Washington, DC 20585

1 S. Rousso, Associate Director
(RW-50)
Office of Contract Business
Management
OCRWM
US Department of Energy
1000 Independence Avenue, SW
Washington, DC 20585

1 T. Wood (RW-52)
Director, M&O Management Division
OCRWM
US Department of Energy
1000 Independence Avenue, SW
Washington, DC 20585

4 Victoria F. Reich, Librarian
Nuclear Waste Technical Review Board
1100 Wilson Blvd, Suite 910
Arlington, VA 22209

5 C. P. Gertz, Project Manager
Yucca Mountain Site
Characterization Project Office
US Department of Energy
P.O. Box 98608--MS 523
Las Vegas, NV 89193-8608

1 C. L. West, Director
Office of External Affairs
DOE Nevada Field Office
US Department of Energy
P.O. Box 98518
Las Vegas, NV 89193-8518

12 Technical Information Officer
DOE Nevada Field Office
US Department of Energy
P.O. Box 98518
Las Vegas, NV 89193-8518

1 P. K. Fitzsimmons, Technical Advisor
Office of Assistant Manager for
Environmental Safety and
Health
DOE Nevada Field Office
US Department of Energy
P.O. Box 98518
Las Vegas, NV 89193-8518

1 D. R. Elle, Director
Environmental Protection and Division
DOE Nevada Field Office
US Department of Energy
P.O. Box 98518
Las Vegas, NV 89193-8518

1 Repository Licensing & Quality
Assurance
Project Directorate
Division of Waste Management
US NRC
Washington, DC 20555

1 Senior Project Manager for Yucca
Mountain
Repository Project Branch
Division of Waste Management
US NRC
Washington, DC 20555

1 NRC Document Control Desk
Division of Waste Management
US NRC
Washington, DC 20555

1 Philip S. Justus
NRC Site Representive
301 E Stewart Avenue, Room 203
Las Vegas, NV 89101

1 E. P. Binnall
Field Systems Group Leader
Building 50B/4235
Lawrence Berkeley Laboratory
Berkeley, CA 94720

1 Center for Nuclear Waste
Regulatory Analyses
6220 Culebra Road
Drawer 28510
San Antonio, TX 78284

3 W. L. Clarke
Technical Project Officer - YMP
Attn: YMP/LRC
Lawrence Livermore National
Laboratory
P.O. Box 5514
Livermore, CA 94551

1 J. A. Blink
Deputy Project Leader
Lawrence Livermore National Laboratory
101 Convention Center Drive
Suite 820, MS 527
Las Vegas, NV 89109

4 J. A. Canepa
Technical Project Officer - YMP
N-5, Mail Stop J521
Los Alamos National Laboratory
P.O. Box 1663
Los Alamos, NM 87545

1 H. N. Kalia
Exploratory Shaft Test Manager
Los Alamos National Laboratory
Mail Stop 527
101 Convention Center Dr., #820
Las Vegas, NV 89101

1 N. Z. Elkins
Deputy Technical Project Officer
Los Alamos National Laboratory
Mail Stop 527
101 Convention Center Dr., #820
Las Vegas, NV 89101

5 L. E. Shephard
Technical Project Officer - YMP
Sandia National Laboratories
Organization 6302
P.O. Box 5800
Albuquerque, NM 87185

1 J. F. Devine
Asst Director of Engineering Geology
US Geological Survey
106 National Center
12201 Sunrise Valley Drive
Reston, VA 22092

1 L. R. Hayes
Technical Project Officer
Yucca Mountain Project Branch
MS 425
US Geological Survey
P.O. Box 25046
Denver, CO 80225

1 V. R. Schneider
Asst. Chief Hydrologist--MS 414
Office of Program Coordination and Technical Support
US Geological Survey
12201 Sunrise Valley Drive
Reston, VA 22092

1 J. S. Stuckless
Geologic Division Coordinator
MS 913
Yucca Mountain Project
US Geological Survey
P.O. Box 25046
Denver, CO 80225

1 D. H. Appel, Chief
Hydrologic Investigations Program
MS 421
US Geological Survey
P.O. Box 25046
Denver, CO 80225

1 E. J. Helley
Branch of Western Regional Geology
MS 427
US Geological Survey
345 Middlefield Road
Menlo Park, CA 94025

1 R. W. Craig, Chief
Nevada Operations Office
US Geological Survey
101 Convention Center Drive
Suite 860, MS 509
Las Vegas, NV 89109

1 D. Zesiger
US Geological Survey
101 Conventional Center Drive
Suite 860, MS 509
Las Vegas, NV 89109

1 G. L. Ducret, Associate Chief
Yucca Mountain Project Division
US Geological Survey
P.O. Box 25046
421 Federal Center
Denver, CO 80225

1 A. L. Flint
US Geological Survey
MS 721
P.O. Box 327
Mercury, NV 89023

1 D. A. Beck
Water Resources Division, USGS
6770 S Paradise Road
Las Vegas, NV 89119

1 P. A. Glancy
US Geological Survey
Federal Building, Room 224
Carson City, NV 89701

1 Sherman S.C. Wu
Branch of Astrogeology
US Geological Survey
2255 N Gemini Drive
Flagstaff, AZ 86001

1 J. H. Sass - USGS
Branch of Tectonophysics
2255 N Gemini Drive
Flagstaff, AZ 86001

1 DeWayne Campbell
Technical Project Officer - YMP
US Bureau of Reclamation
Code D-3790
P.O. Box 25007
Denver, CO 80225

1 J. M. LaMonaca
Records Specialist
US Geological Survey
421 Federal Center
P.O. Box 25046
Denver, CO 80225

1 W. R. Keefer - USGS
913 Federal Center
P.O. Box 25046
Denver, CO 80225

1 M. D. Voegele
Technical Project Officer - YMP
SAIC
101 Convention Center Drive
Suite 407
Las Vegas, NV 89109

2 L. D. Foust
Nevada Site Manager
TRW Environmental Safety Systems
101 Convention Center Drive
Suite 540, MS 423
Las Vegas, NV 89109

1 C. E. Ezra
YMP Support Office Manager
EG&G Energy Measurements Inc
MS V-02
P.O. Box 1912
Las Vegas, NV 89125

1 E. L. Snow, Program Manager
Roy F. Weston Inc
955 L'Enfant Plaza SW
Washington, DC 20024

1 Technical Information Center
Roy F. Weston Inc
955 L'Enfant Plaza SW
Washington, DC 20024

1 D. Hedges, Vice President, QA
Roy F. Weston Inc
4425 Spring Mountain Road
Suite 300
Las Vegas, NV 89102

1 D. L. Fraser, General Manager
Reynolds Electrical & Engineering Co, Inc
MS 408
P.O. Box 98521
Las Vegas, NV 89193-8521

1 B. W. Colston, President and General Manager
Las Vegas Branch
Raytheon Services Nevada
MS 416
P.O. Box 95487
Las Vegas, NV 89193-5487

1 R. L. Bullock
Technical Project Officer - YMP
Raytheon Services Nevada
Suite P-250, MS 403
101 Convention Center Drive
Las Vegas, NV 89109

1 Paul Eslinger, Manager
PASS Program
Pacific Northwest Laboratories
P.O. Box 999
Richland, WA 99352

1 A. T. Tamura
Science and Technology Division
OSTI
US Department of Energy
P.O. Box 62
Oak Ridge, TN 37831

1 Carlos G. Bell Jr
Professor of Civil Engineering
Civil and Mechanical Engineering Dept.
University of Nevada, Las Vegas
4505 S Maryland Parkway
Las Vegas, NV 89154

1 P. J. Weeden, Acting Director
Nuclear Radiation Assessment Div.
US EPA
Environmental Monitoring
Systems Lab
P.O. Box 93478
Las Vegas, NV 89193-3478

1 ONWI Library
Battelle Columbus Laboratory
Office of Nuclear Waste Isolation
505 King Avenue
Columbus, OH 43201

1 T. Hay, Executive Assistant
Office of the Governor
State of Nevada
Capitol Complex
Carson City, NV 89710

3 R. R. Loux
Executive Director
Agency for Nuclear Projects
State of Nevada
Evergreen Center, Suite 252
1802 N. Carson Street
Carson City, NV 89710

1 C. H. Johnson
Technical Program Manager
Agency for Nuclear Projects
State of Nevada
Evergreen Center, Suite 252
1802 N. Carson Street
Carson City, NV 89710

1 John Fordham
Water Resources Center
Desert Research Institute
P.O. Box 60220
Reno, NV 89506

1 David Rhode
Desert Research Institute
P.O. Box 60220
Reno, NV 89506

1 Eric Anderson
Mountain West Research-
Southwest Inc
2901 N Central Avenue #1000
Phoenix, AZ 85012-2730

1 The Honorable Cyril Schank
Chairman
Churchill County Board of
Commissioners
190 W First Street
Fallon, NV 89406

1 Dennis Bechtel, Coordinator
Nuclear Waste Division
Clark County Department of
Comprehensive Planning
301 E Clark Avenue, Suite 570
Las Vegas, NV 89101

1 Juanita D. Hayes
Nuclear Waste Repository
Oversight Program
Esmeralda County
P.O. Box 490
Goldfield, NV 89013

1 Yucca Mountain Information
Office
Eureka County
P.O. Box 714
Eureka, NV 89316

1 Brad Mettam
Inyo County Yucca Mountain
Repository Assessment Office
Drawer L
Independence, CA 93526

1 The Honorable Gloria Derby
Chairman
Lander County Board of
Commissioners
315 South Humbolt
Battle Mountain, NV 89820

1 The Honorable Edward E. Wright
Chairman
Lincoln County Board of
Commissioners
P.O. Box 90
Pioche, NV 89043

1 Vernon E. Poe
Office of Nuclear Projects
Mineral County
P.O. Box 1026
Hawthorne, NV 89415

1 The Honorable Barbara J. Raper
Chairman
Nye County Board of
Commissioners
P.O. Box 1240
Pahrump, NV 89041

1 Planning Department
Nye County
P.O. Box 153
Tonopah, NV 89049

1 Florindo Mariani
White Pine County Nuclear
Waste Project Office
457 Fifth Street
Ely, NV 89301

5 Judy Foremaster
City of Caliente Nuclear Waste
Project Office
P.O. Box 158
Caliente, NV 89008

1 Jason Pitts
Lincoln County Nuclear Waste
Project Office
Lincoln County Courthouse
Pioche, NV 89043

1 Economic Development Dept.
City of Las Vegas
400 E. Stewart Avenue
Las Vegas, NV 89101

1 Commmunity Planning and
Development
City of North Las Vegas
P.O. Box 4086
North Las Vegas, NV 89030

1 Community Development and
Planning
City of Boulder City
P.O. Box 61350
Boulder City, NV 89006

1 Commission of the European
Communities
200 Rue de la Loi
B-1049 Brussells
BELGIUM

2 M. J. Dorsey, Librarian
YMP Research and Study Center
Reynolds Electrical &
Engineering Co Inc
MS 407
P.O. Box 98521
Las Vegas, NV 89193-8521

1 Amy Anderson
Argonne National Laboratory
Building 362
9700 S Cass AVenue
Argonne, IL 60439

1 Steve Bradhurst
P.O. Box 1510
Reno, NV 89505

1 Michael L. Baughman
35 Clark Road
Fiskdale, MA 01518

1 Glenn Van Roekel
Director of Community
Development
City of Caliente
P.O. Box 158
Caliente, NV 89008

1 Ray Williams, Jr
P.O. Box 10
Austin, NV 89310

1 R. F. Pritchett
Technical Project Officer - YMP
Reynolds Electrical &
Engineering Co Inc
MS 408
P.O. Box 98521
Las Vegas, NV 89193-8521

1 Charles Thistlethwaite, AICP
Associate Planner
Inyo County Planning Department
Drawer L
Independence, CA 93526

1 Les Bradshaw
Nye County District Attorney
P.O. Box 593
Tonopah, NV 89049

1 Dr. Moses Karakouzian
1751 E Reno #125
Las Vegas, NV 89119

1 Dr. R. T. Allen
Pacifica Technology
P.O. Box 148
Del Mar, CA 92014

1 Prof. S. Atluri
Center for the Advancement of
Computational Mechanics
School of Civil Engineering
Georgia Institute of Technology
Atlanta, GA 30332

1 Dr. Ali S. Argon
Department of Mechanical
Engineering
MIT
Cambridge, MA 02139

1 Dr. William E. Bachrach
Aerojet Research Propulsion Inst.
P. O. Box 13502
Sacramento, CA 95853-4502

1 Mr. Ken Bannister
USA Ballistic Research Lab
SLCBR-IB-M
Aberdeen Proving Grounds, MD
21005-5066

1 Prof. E. B. Becker
Department of Aerospace
Engineering and
Engineering Mechanics
University of Texas at Austin
Austin, TX 78712-1085

1 Prof. M. Stern
Department of Aerospace
Engineering and
Engineering Mechanics
University of Texas at Austin
Austin, TX 78712-1085

1 Prof. J. T. Oden
Department of Aerospace Engineering
and Engineering Mechanics
University of Texas at Austin
Austin, TX 78712-1085

1 Prof. T. Belytschko
Department of Civil Engineering
Northwestern University
Evanston, IL 60201

1 Dr. Kenneth W. Brown
Computer Aided Engineering
Associates, Inc.
398 Old Sherman Hill Rd.
Woodbury, CT 06798

1 Dr. Tom Canfield
Argonne National Laboratories
9700 S. Cass Ave CTD/221
Argonne, IL 60439-4844

1 Mr. Tien S. Chou
EG&G Mound
P.O. Box 3000
Miamisburg, OH 45343

1 Mr. Chuck Charman
GA Technologies
P.O. Box 81608
San Diego, CA 92138

1 Mr. Ken K. Chipley
Martin Marietta Energy Systems
P.O. Box 2009
Oak Ridge, TN 37831-8053

1 Mr. Ken P. Chong
Dept. of Civil Engineering
University of Wyoming
Laramie, WY 82071

1 Dr. S. C. (Tony) Chou
U.S. Army Materials Technology Lab
SLCMT-BM
Watertown, MA 02172-0001

1 Mr. Dwight Clark
Mail Stop 281
Morton Thiokol Corp.
P. O. Box 524
Brigham City, UT 84302

1 Mr. Gerald Collingwood
Morton Thiokol, Inc.
Huntsville, AL 35807-7501

1 Steven Crouch
GeoLogic Research, Inc.
1313 Fifth St. SE, Suite 226
Minneapolis, MN 55414

1 Dr. Ian Cullis
XTZ Division
Royal Armament R&D Establishment
Fort Halstead
Sevenoaks, Kent
United Kingdom

1 Mr. Richard E. Danell
Research Officer
Central Research Laboratories
BHP Research/New Technology
P.O. Box 188
Wallsend NSW 2287
Australia

1 Mr. William A. Danne
Strategic Systems Division, MS#50
Teledyne Brown Engineering
P.O. Box 07007
Huntsville, AL 35807-7007

1 Dr. C. S. Desai
Dept. of Civil Eng. & Eng. Mech.
University of Arizona
Tucson, AZ 85721

1 Mr. Ramji Digumarthi
Org. 8111, Bldg. 157
Lockheed MSD
P.O. Box 3504
Sunnyvale, CA 94088-3504

1 Mr. J. Donald Dixon
Spokane Research Center
U.S. Bureau of Mines
315 Montgomery Avenue
Spokane, WA 99207-2291

1 Dr. Hamid Maleki
Spokane Research Center
U.S. Bureau of Mines
315 Montgomery Avenue
Spokane, WA 99207-2291

1 Prof. Robert Dodds, Jr.
Department of Civil Engineering
3140 Newmark Laboratory, MC-250
University of Illinois at Urbana
Urbana, IL 61801-2397

1 Dr. R. S. Dunham
Anatech International Corp.
3344 N. Torrey Pines Ct.
Suite 320
La Jolla, CA 92037

1 Dr. Joe Rashid
Anatech International Corp.
3344 N. Torrey Pines Ct.
Suite 320
La Jolla, CA 92037

1 Dr. Arlo Fossum
RE/SPEC Inc.
Box 725
Rapid City, SD 57709

1 Dr. Russel Garnsworthy
CRA Advanced Tech Development
G.P.O. Box 384D
Melbourne 3001, Australia

1 Loren K. Miller
Goodyear Technical Center
P.O. Box 3531
Akron, OH 44309-3531

1 David Wismer, D/410F
Goodyear Technical Center
P.O. Box 3531
Akron, OH 44309-3531

1 Dr. Gerry Goudreau
Methods Development Group
Mechanical Engineering Department
Lawrence Livermore National Lab
Livermore, CA 94550

1 Mr. H. L. Hassenpflug
B&W Fuel Company
P.O. Box 10935
3315 Old Forest Rd
Lynchburg, VA 24501

1 Dr. David Hibbitt
Hibbitt, Karlsson & Sorrensen, Inc.
100 Medway St.
Providence, RI 02906

1 Dr. Joop Nagtegaal
Hibbitt, Karlsson & Sorrensen, Inc.
100 Medway St.
Providence, RI 02906

1 Dr. D. P. Flanagan
Hibbitt, Karlsson & Sorrensen, Inc.
100 Medway St.
Providence, RI 02906

1 Dr. L. M. Taylor
Hibbitt, Karlsson & Sorrensen, Inc.
100 Medway St.
Providence, RI 02906

1 Dr. W. C. Mills-Curran
Hibbitt, Karlsson & Sorrensen, Inc.
100 Medway St.
Providence, RI 02906

1 Mr. Richard Hilson
M/S 4G09
GTE Government Systems Corporation
P.O. Box 7188
Mountain View, CA 94039

1 Douglas Holzhauer
Rome Air Development Center
Griffith AFB, NY 13441

1 Dr. William Hufferd
United Technologies
Chemical Systems Division
P.O. Box 50015
San Jose, CA 95150-0015

1 Prof. T. J. R. Hughes
Department of Mechanical Engineering
Stanford University
Palo Alto, CA 94306

1 Mr. James P. Johnson
Rm L120, CPC Analysis Dept.
General Motors Corp.
Engineering Center
30003 Van Dyke Avenue
Warren, MI 48090-9060

1 Mr. Jerome B. Johnson
USACRREL
Building 4070
Ft. Wainwright, AK 99703

1 Ken Johnson
Theoretical and Applied Mechanics
Group
Battelle Pacific Northwest Laboratories
P.O. Box 999
Richland, WA 99352

1 Dr. Gordon R. Johnson
Honeywell, Inc.
5901 S. County Rd. 18
Edina, MN 55436

1 Dr. Rembert Jones
Office of Naval Research
Structural Mechanics Div. (Code 434)
800 N. Quincy Street
Arlington, VA 22217

1 Dr. Alan S. Kushner
Office of Naval Research
Structural Mechanics Div. (Code 434)
800 N. Quincy Street
Arlington, VA 22217

1 Dr. Mike Katona
TRW Ballistic Missiles Division
Bldg 527, Rm 709
P.O. Box 1310
San Bernadino, CA 92402

1 Mr. Richard Lung
TRW Ballistic Missiles Division
Bldg 527, Rm 709
P.O. Box 1310
San Bernadino, CA 92402

1 Mr. Gary Ketner
Applied Mechanics and Structures
Battelle Pacific Northwest Laboratories
P.O. Box 999
Richland, WA 99352

1 Dr. Sam Key
RE/SPEC Inc.
4775 Indian School NE, Suite 300
Albuquerque, NM 87110-3827

1 Prof. Raymond D. Krieg
Engineering Science and Mechanics
301 Perkins Hall
University of Tennessee
Knoxville, TN 37996-2030

1 Don D. Kunard
Analytical Systems Engineering Corp.
1725 Jefferson Davis Hwy, Suite 212
Arlington, VA 22202

1 Prof. J. K. Lee
Department of Engineering Mechanics
Ohio State University
Columbus, OH 43210

1 Mr. Brett Lewis
APTEK
1257 Lake Plaza Drive
Colorado Springs, CO 80906-3578

1 J. Hopson, T3, MS B216
Los Alamos National Laboratory
Los Alamos, NM 87545

1 R. Hill, P15, MS D449
Los Alamos National Laboratory
Los Alamos, NM 87545

1 F. Guerra, WX-11, MS C931
Los Alamos National Laboratory
Los Alamos, NM 87545

1 B. M. Wheat, MEE-4, MS G787
Los Alamos National Laboratory
Los Alamos, NM 87545

1 E. S. Idar, MEE-4, MS G787
Los Alamos National Laboratory
Los Alamos, NM 87545

1 M. W. Lewis, MEE-4, MS G787
Los Alamos National Laboratory
Los Alamos, NM 87545

1 J. P. Hill, WX-11, MS C931
Los Alamos National Laboratory
Los Alamos, NM 87545

1 D. J. Sandstorm, MST-DO, MS G756
Los Alamos National Laboratory
Los Alamos, NM 87545

1 K. A. Meyer, X-3 MS F663
Los Alamos National Laboratory
Los Alamos, NM 87545

1 W. A. Cook, N-6, MS K557
Los Alamos National Laboratory
Los Alamos, NM 87545

1 P. T. Maulden, N-6, MS K557
Los Alamos National Laboratory
Los Alamos, NM 87545

1 J. J. Ruminer, WX-11, MS C931
Los Alamos National Laboratory
Los Alamos, NM 87545

1 S. P. Girrens, MEE-13, MS J576
Los Alamos National Laboratory
Los Alamos, NM 87545

1 Trent R. Logan
Rockwell International Corp.
P.O. Box 92098
Los Angeles, CA 90009

1 J. L. Fales, MEE-13, MS J575
Los Alamos National Laboratory
Los Alamos, NM 87545

1 J. D. Allen, MEE-4, MS G787
Los Alamos National Laboratory
Los Alamos, NM 87545

1 D. A. Rabern, MEE-4, MS G787
Los Alamos National Laboratory
Los Alamos, NM 87545

1 M. W. Burkett, MEE-4, MS G787
Los Alamos National Laboratory
Los Alamos, NM 87545

1 J. H. Fu, MEE-4, MS G787
Los Alamos National Laboratory
Los Alamos, NM 87545

1 P. R. Romero, MEE-4, MS G787
Los Alamos National Laboratory
Los Alamos, NM 87545

1 P. S. Follansbee, MST-DO, MS G756
Los Alamos National Laboratory
Los Alamos, NM 87545

1 R. F. Davidson, N-6, MS K557
Los Alamos National Laboratory
Los Alamos, NM 87545

1 D. Mandell, X-3, MS F663
Los Alamos National Laboratory
Los Alamos, NM 87545

1 J. N. Johnson, N-6, MS K557
Los Alamos National Laboratory
Los Alamos, NM 87545

1 J. K. Dienes, N-6, MS K557
Los Alamos National Laboratory
Los Alamos, NM 87545

1 S. Marsh, N-6, MS K557
Los Alamos National Laboratory
Los Alamos, NM 87545

1 L. H. Sullivan, N-6, MS K557
Los Alamos National Laboratory
Los Alamos, NM 87545

1 R. B. Parker, MEE-4, MS G787
Los Alamos National Laboratory
Los Alamos, NM 87545

1 D. C. Nelson, MEE-4, MS G787
Los Alamos National Laboratory
Los Alamos, NM 87545

1 T. A. Butler, MEE-13, MS J576
Los Alamos National Laboratory
Los Alamos, NM 87545

1 J. G. Bennett, MEE-13, MS J576
Los Alamos National Laboratory
Los Alamos, NM 87545

1 C. A. Anderson, MEE-13, MS J576
Los Alamos National Laboratory
Los Alamos, NM 87545

1 D. L. Jaeger, WX-11, MS K557
Los Alamos National Laboratory
Los Alamos, NM 87545

1 David Medina
PL/WSSD
Phillips Laboratory (AFSC)
Kirtland AFB, NM 87117-6008

1 Mr. Hans Mair, Code R14
Naval Surface Warfare Center
10901 New Hampshire Ave.
Silver Spring, MD 20903-5000

1 Mr. Andrew Wardlaw, Jr., Code R44
Naval Surface Warfare Center
10901 New Hampsure Ave.
Silver Spring, MD 20903-5000

1 Dr. Jack Maison
Engineering Cybernetics, Inc.
1856 Lockhill Selma Rd, Suite 105
San Antonio, TX 78213

1 Joseph Marti
Hamilton Standard Division
of United Technologies
M/S 1-3-BC52
One Hamilton Road
Windsor Locks, CT 06096-1010

1 Darin McKinnis
NASA Pyrotechnics Group, MS EP5
LBJ Space Center
Houston, TX 77058

1 Mr. Craig Miller
Unit 973
Neutron Devices Department
General Electric Company
P.O. Box 2908
Largo, FL 34294-2908

1 Mr. J. J. Murphy, 59-22 B/580
Lockheed Missiles and Space Co.
P. O. Box 3504
Sunnyvale, CA 94088-3504

1 Brian M. Cuthbert, 81-12 B/157
Lockheed Missiles and Space Co.
P. O. Box 3504
Sunnyvale, CA 94088-3504

1 Prof. V. D. Murty
School of Engineering
University of Portland
5000 N. Willamette Blvd.
Portland, OR 97203

1 Prof. S. Nemat-Nasser
Department of Applied Mechanics
and Engineering Sciences
University of California San Diego
La Jolla, CA 92093

1 Prof. Dave Benson
Department of Applied Mechanics
and Engineering Sciences
University of California San Diego
La Jolla, CA 92093

1 Mr. Jim Nemes, Code 6331
Naval Research Lab
Washington, DC 20375-5000

1 Dr. Carl Dyka, Code 6380
Naval Research Lab
Washington, DC 20375-5000

1 Dr. R. E. Nickell
c/o Anatech International Corp.
3344 N. Torrey Pines Ct, Suite 320
La Jolla, CA 92037

1 Mr. Dean Norman
Waterways Experiment Station
P.O. Box 631
Vicksburg, MS 39180

1 Dr. Robert Pardue
Martin Marietta, MS 2
Y-12 Plant, Bldg. 9998
Oak Ridge, TN 37831

1 Dr. T. Kim Parnell
Failure Analysis Associates, Inc.
P.O. Box 3015
Menlo Park, CA 94025

1 Mr. Mitchell R. Phillabaum
Monsanto Research Corp.
MRC-MOUND
Miamisburg, OH 45342

1 John Cowles, MS 118-38
Pratt & Whitney Aircraft
400 Main St.
East Hartford, CT 06108

1 Mick Bruskotter, MS 114-38
Pratt & Whitney Aircraft
400 Main St.
East Hartford, CT 06108

1 Dr. Harold E. Read
S-Cubed
P.O. Box 1620
La Jolla, CA 92038-1620

1 Dr. Douglas Reeder
Hardening Technology Dept.
General Research Corp.
P.O. Box 6770
Santa Barbara, CA 93160-6770

1 Dr. J. A. Reuscher
Texas A & M
Dept. of Nuclear Engineering
College Station, Texas 77843

1 Mr. J. S. (Gus) Rice
Caterpillar Inc. Technical Center
Division 927
P.O. Box 1875
Peoria, IL 61656-1875

1 Steven Rieco
POD Associates, Inc.
2309 Renard Pl, Suite 201
Albuquerque, NM 87106

1 Mr. Samit Roy
Dept. of Engineering Mechanics
Southwest Research Institute
P.O. Drawer 28510
San Antonio, TX 78284

1 Mr. Donald W. Sandidge
AMSMI-RLA
U.S. Army Missile Command
Redstone Arsenal, AZ 35898-5247

1 Mr. Steven Sauer
Ktech Corporation
901 Pennsylvania Ave NE
Albuquerque, NM 87110

1 Martin Schmidt, M/S 4G09
WL/MNSA
Eglin AFB, FL 32542-5434

1 Mr. Luka Serdar, Jr.
Kaman Sciences Corporation
AviDyne Office
83 Second Ave
Burlington, MA 01803-4479

1 Harvey Singer
Science Applications International
P.O. Box 1303
McLean, VA 22102-13031

1 Mark E. Smith
Arvin Calspan Corp.
AEDC Division, M/S 440
Arnold AFB, TN 37389-9998

1 Mr. Ray Stoudt
Lawrence Livermore National Lab
P.O. Box 808, L200
Livermore, CA 94550

1 Prof. D. V. Swenson
Mechanical Engineering Department
Kansas State University
Manhattan, KS 66506

1 Mr. David L. Conover
Swanson Analysis Systems, Inc.
P. O. Box 65
Houston, PA 15342-0065

1 Mr. James W. Jones
Swanson Service Corporation
18700 Beach Blvd.
Suite 200-210
Huntington Beach, CA 92648

1 Mr. Sing C. Tang
Rm 3039 Scientific Lab
P. O. Box 2053
Dearborn, MI 48121-2053

1 Dave Edwards, MS 129-13
United Technologies Research Center
411 Silver Lane
East Hartford, CT 06108

1 Robert LaBarre, MS 129-20
United Technologies Research Center
411 Silver Lane
East Hartford, CT 06108

1 Tony Giamei, MS 129-22
United Technologies Research Center
411 Silver Lane
East Hartford, CT 06108

1 Tom Vasko
United Technologies Research Center
411 Silver Lane
East Hartford, CT 06108

1 Mr. David Wade, 36E
Bettis Atomic Power Laboratory
P.O. Box 79
West Mifflland, PA 15122

1 Dr. Krishan K. Wahi
Gram, Inc.
1709 Moon NE
Albuquerque, NM 87112

1 Dr. Paul T. Wang
Fabricating Technology Division
Alcoa Technical Center
Alcoa Center, PA 15069

1 Dr. Ted B. Wertheimer
MARC Analysis Research Corporation
260 Sheridan Ave, Suite 309
Palo Alto, CA 94306

1 Todd Hoover
Westinghouse Electric Coporation
Bettis Atomic Power Laboratory
P.O. Box 79
West Mifflin, PA 15122-0079

1 Claire Knolle
Westinghouse Electric Coporation
Bettis Atomic Power Laboratory
P.O. Box 79
West Mifflin, PA 15122-0079

1 Dan Kotcher
Westinghouse Electric Corporation
Bettis Atomic Power Laboratory
P.O. Box 79
West Mifflin, PA 15122-0079

1 Wayne Long
Westinghouse Electric Coporation
Bettis Atomic Power Laboratory
P.O. Box 79
West Mifflin, PA 15122-0079

1 Prof. Tomasz Wierzbicki
Dept. of Ocean Engineering
MIT
Cambridge, MA 02139

1 Dr. John Wilson
Department of Geoscience
NM Institute of Mining & Technology
Socorro, NM 87801

1 Dr. Albert Yao
SMCRI-SEE-A
Rock Island Arsenal
Rock Island, IL 61299-5000

1 Mr. J. A. Zukas
Computational Mechanics Consultants
8600 La Salle Road, Suite 614
Towson, MD 21204

SANDIA INTERNAL

1	1200	George Allshouse
1	1231	Thomas Sanford
1	1239	Frank Dempsey
1	1400	Ed Barsis
10	1425	Johnny Biffle
1	1425	Stephen Attaway
20	1425	Mark Blanford
1	1425	William Bohnhoff
30	1425	Marilyn Smith
1	1431	Michael McGlaun
1	1431	James Peery
1	1434	David Martinez
1	1501	Carl Peterson
1	1502	Paul Hommert
1	1561	Harold Morgan
1	1561	Edward Hoffman
1	1561	Gerald Wellman
1	1561	James Koteras
1	1561	Joe Weatherby
1	1561	Jose Arguello Jr.
1	1561	Mark Rashid
1	1561	Martin Heinstein
1	1561	Michael Neilsen
1	1561	Robert Chambers
1	1561	Steven Burchett
1	1561	Vicki Bergmann
1	1562	Robert Thomas
1	1562	Gregory Sjaardema
1	1562	Roy Johnson
1	1562	Peter Stirbis
1	1832	Jeanne Ramage
1	1912	William Mason
1	2565	Stephen Montgomery
1	2814	Randall Lober
1	5600	Dennis Hayes
1	5941	John Schamaun
1	6000	Dan Hartley
1	6112	Dale Preece
1	6112	Norman Warpinski
1	6113	Stephen Bauer
1	6113	Brian Ehgartner
1	6313	John Holland
1	6313	Joseph Jung
1	6313	John Pott
1	6313	Alex Treadway
1	6316	Mike Wernig
1	6500	James Rice
1	6514	Jim Fisk
1	6515	Mike Rightley
1	6522	Joel Miller
1	6642	Douglas Ammerman
1	8702	Bill Robinson
1	8712	Kim Mahin
1	8240	George Johnson
1	8741	Michael Chiesa
1	8742	Bruce Kistler
1	8743	Melvin Callabresi
1	8743	Douglas Bammann
1	8743	Juanita Benson
1	8743	Lee Bertram
1	8743	Mark Horstemeyer
1	8743	James Lathrop
1	8744	Arthur Ortega
1	8745	William Winters
1	6300	D.E. Miller
1	6302	L.E. Shephard
1	6312	F.W. Bingham
6	6313	L.S. Costin
2	6352	G.M. Gerstner-Miller for 100/124231/SAND87-1305/NQ
2	6352	G.M. Gerstner-Miller for DRMS files
1	6319	R.R. Richards
3	6115	P. B. Davies
20	6352	WMT Library
1	6410	D.A. Dahlgren
5	7141	Technical Library
1	7151	Technical Publications
10	7613-2	Document Processing for DOE/OSTI
1	8523-2	Central Technical Files

★ U.S. GOVERNMENT PRINTING OFFICE 1992—774-122/60114

THIS PAGE INTENTIONALLY LEFT BLANK

SAND87-1305

The number in the lower right-hand corner is an accession number used for Office of Civilian Radioactive Waste Management purposes only. It should not be used when ordering this publication.

NNA.930305.0104